Univerzitet u Beogradu
Matematički fakultet

Master rad

Uslovna slučajna polja i primena na rastavljanje reči na kraju reda u srpskom jeziku

Student: Ivana Medić 1124/2013
Mentor: Prof. dr Gordana Pavlović-Lažetić
Članovi komisije: Prof. dr Duško Vitas, Doc. dr Mladen Nikolić

Sadržaj

1. Uvod 3
2. Uslovna slučajna polja 4
    2.1 Grafički modeli 5
    2.2 Diskriminativni modeli 7
        2.2.1 Linearna regresija 7
        2.2.2 Logistička regresija 8
        2.2.3 Klasifikacija pomoću logističke reresije 8
        2.2.4 Provera parametara i princip maksimalne verodostojnosti 10
        2.2.5 Model maksimalne entropije 12
    2.3 Skriveni Markovljevi modeli 13
    2.4 Markovljevi modeli maksimalne entropije 17
    2.5 Linearna uslovna slučajna polja 19
    2.6 Generalna uslovna slučajna polja 22
    2.7 Algoritmi 24
        2.7.1 Pregled algoritama 24
        2.7.2 Viterbi i unapred-unazad algoritmi za linearna uslovna slučajna polja 24
        2.7.3 Zaključivanje kod generalnih uslovnih slučajnih polja 25
    2.8 Obučavanje parametara 28
        2.8.1 Slučaj linearnih uslovnih slučajnih polja 28
        2.8.2 Metoda stohastičkog gradijentnog spusta 30
        2.8.3 Slučaj generalnih uslovnih slučajnih polja 31
3. Rastavljanje reči na kraju reda 32
    3.1 Istorijat 32
    3.2 Pravila za rastavljanje reči na kraju reda u srpskom jeziku 33
4. Primena uslovnih slučajnih polja na rastavljanje reči na kraju reda u srpskom jeziku 36
    4.1 Kodiranje promenljivih 36
    4.2 Karakteristične funkcije 36
    4.3 Skup za obučavanje i skup za testiranje 39
    4.4 CRF++ 39
    4.5 Implementacija 43
    4.6 Rezultati 44
5. Zaključak 46
6. Reference 47

1. Uvod

Mašinsko učenje je grana računarstva kojoj se opravdano pridaje veliko interesovanje. Računari su svoje mesto u nauci i svakodnevnom životu zauzeli zahvaljujući mogućnosti da urade baš ono što im je rečeno, neuporedivo brže od čoveka. Međutim, vremenom smo poželeli da budu "pametniji", da prepoznaju lica i slova, pronađu imena ogranizacija u vestima ili znaju da odgovore na postavljeno pitanje. Kako bi to bilo omogućeno, razvijeno je više metoda mašinskog učenja. Jedna od njih, posebno pogodna za procesiranje prirodnih jezika, su uslovna slučajna polja, tema ovog rada.

U poglavlju 2.1 uvodi se specifična vrsta neusmerenog grafa kojim se definišu uslovna slučajna polja. Posebna pažnja posvećena je linearnim uslovnim slučajnim poljima, kao njihovoj najjednostavnijoj ali i najznačajnijoj varijanti. Ona se može posmatrati kao diskriminativni analogon skrivenih Markovljevih modela, ili kao nadogradnja na logističku regresiju. Ova dva probabilistička modela su ukratko opisana u poglavljima 2.2 i 2.3 radi boljeg razumevanja porekla uslovnih slučajnih polja. Motivacija za ovu metodu je premošćavanje problema koji se javljao u do tada razvijenim modelima, pre svega Markovljevim modelima maksimalne entropije, o čemu će takođe biti reči u poglavlju 2.4. U poglavljima 2.5 i 2.6 uvode se linearna i generalna uslovna slučajna polja, dok se poglavlja 2.7 i 2.8 bave algoritmima vezanim za ove metode.

Poglavlje 3 opisuje problem rastavljanja reči na kraju reda. U pitanju je problem na koji se svakodnevno nailazi pri pripremi teksta za štampu ili prikaz. Poglavlje počinje kratkom istorijom problema i pravilima koje ispravna podela treba da zadovolji.

Konačno, u poglavlju 4 razmatra se primena uslovnih slučajnih polja na konkretan problem. Predstavljeno je kodiranje sekvenci, opisane su karakteristične funckije i parametri modela i skupovi za testiranje i obučavanje. Na konstruisan model primenjen je softver CRF++ koji je izvršio učenje parametara i predikciju. Rezultati su analizirani u poglavlju 4.6.

2. Uslovna slučajna polja

Uslovna slučajna polja predstavljena su 2001. godine kao novi probabilistički model za segmentiranje i tagovanje sekvencijalnih podataka [1]. Taj problem nije bio nov, i različite metode za njegovo rešavanje su razvijene decenijama pre toga. Međutim, bio je potreban efikasan model koji bi omogućio posmatranje više karakterika ulaznih podataka, i pritom prevazišao problem naklonosti tagovima (engl. *label bias problem*) [2], o čemu će biti više reči u nastavku.

CRF je grafovski diskriminativni probabilistički model mašinskog učenja za struktuiranu predikciju. Pre svega, trebalo bi razjasniti ove pojmove. Često imamo neke podatke i njihove karakteristike, na osnovu kojih želimo da zaključimo nešto novo. Ovo može biti problem određivanja vrste reči u tekstu, prepoznavanje govora, ili, što je tema ovog rada, rastavljanje reči na kraju reda. Uglavnom se radi o sekvencijalnim podacima, koji su međusobno povezani, kao što se slučaj sa tekstom. Radi preglednosti, uvedimo odmah oznake koje će biti korišćenje u ostatku rada. Neka je dat vektor $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_T\}$, gde je T dužina sekvence, a svaki $\boldsymbol{x}_i$ predstavlja vektor karakteristika podatka na poziciji i. Za svaki od tih vektora, potrebno je odrediti tag $\boldsymbol{y}_i$. U slučaju određivanja vrste reči, T bi bio broj reči u tekstu, $\boldsymbol{y}_i$ vrsta reči na poziciji i, a svaki $\boldsymbol{x}_i$ bi bio vektor informacija o i-toj reči kojima se raspolaže - sama reč, informacije o prefiksima, sufiksima, kapitalizaciji... Kako bi bilo kakva predikcija bila moguća, potrebno je model snabdeti informacijama na osnovu kojih će on dalje učiti. Skup takvih podataka $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})$ naziva se skup za obučavanje. U ovom radu se podrazumeva da on u sebi poseduje korektne tagove za sve sekvence, te se radi o nadgledanom učenju. Nakon što model nauči da generiše tagove na osnovu skupa za obučavanje, može se evaluirati na nezavisnom, takođe unapred generisnom, skupu za testiranje. Ovaj korak je bitan jer se od modela očekuje da predviđa tagove za podatke koje nije video u procesu učenja, u čemu leži osnova mašinskog učenja.

Problem predikcije se svodi na pronalaženje raspodela verovatnoća $p(\boldsymbol{y}|\boldsymbol{x})$. Ovom problemu se može pristupiti na dva načina. Jedan pristup ovom je da se tražena verovatnoća dobija na osnovu zajedničke verovatnoće $p(\boldsymbol{x}, \boldsymbol{y})$ i raspodela $p(\boldsymbol{x})$, i radi se o generativnim probabilističkim modelima. Sa druge strane, diskriminativni modeli posmatraju traženu uslovnu verovatnoću direktno, što znači da nije neophodno modelirati promenljive $\boldsymbol{x}$, čija nam raspodela verovatnoća u mnogim problemima i nije od značaja, dok se omogućuje predikcija vrednosti promenljivih $\boldsymbol{y}$ [3]. Ovo povlači mogućnost uključenja velikog broja karakteristika, bez većeg uticaja na složenost predikcije. Za svaki diskriminativni model postoji njegov generativni analogon, i obrnuto. U ovom smislu, linearna uslovna slučajna polja su analogna skrivenim Markovljevim modelima, a logistička regresija, čija su linearna

uslovna slučajna polja generalizacija, analogna je naivnom Bajesovom modelu. Više o ovim pristupima, i načinu na koji se oni mogu kombinovati, može se naći u [3].

Kada je potrebno predstaviti problem sa velikim brojem promenljivih, što je uglavnom slučaj sa problemima mašinskog učenja, grafovi su se pokazali kao veoma pogodna reprezentacija. Uslovna slučajna polja koriste ovu pogodnost pri modeliranju zavisnosti među promenljivim. U linearnom slučaju dovoljno je koristiti listu, dok se najopštija uslovna slučajna polja predstavljaju specifičnim faktor grafovima, koji će detaljnije biti opisani u nastavku poglavlja.

Veći deo ovog poglavlja je posvećen linearnim uslovnim slučajnim poljima. Kao što je već rečeno, on se može posmatrati kao nadogradnja na logističku regresiju, diskriminativni naslednik slučajnih Markovljevih polja, ili poboljšanje Markovljevih polja maksimalne entropije. Dakle, za njihovo jasno razumevanje potrebno je makar osnovno poznavanje ovih modela, čime se ukratko bave poglavlja 2.2, 2.3 i 2.4.

2.1 Grafički modeli

Probabilistički modeli se često predstavljaju grafovima. Oni omogućavaju preglednu reprezentaciju slučajnih promenljivih u modelu, kao i njihove međusobne zavisnosti. U nastavku sledi opis dva tipa ovakvih grafova, preuzet iz [4].

I generativni i diskriminativni modeli kosriste pogodnost grafova, pri čemu se prvi uglavnom predstavljaju usmerenim, a drugi neusmerenim vrstama. Jedan tip neusmerenog grafa koji se često koristi je Markovljeva mreža - graf nad promenljivima u modelu čije su zavisnosti ilustrovane granama. Neka je G taj graf. Ako je s jedna promenljiva, označimo sa $N(s)$ njene susede u G. Kažemo da je raspodela p Markovljeva u odnosu na G ako zadovoljava lokalno Markovljevo svojstvo: za svake dve promenljive V_t i V_s važi

$$p(V_s|V_t, N(s)) = p(V_s|N(s)) \quad . \tag{1}$$

Drugim rečima, promenljiva je uslovljena samo svojim susedima. Ova reprezentacija ipak nije povoljna za uslovna slučajna polja jer se odnosi među promenljivima mogu videli višeznačno. Uslovna slučajna polja definišu se preko faktor grafova.

Neka je dat skup diskretnih slučajnih promenljivih $V = \{V_1, V_2, \ldots V_m\}$. Neka svaka od tih promenljivih, V_s, uzima vrednosti iz skupa W. Dodeljivanje vrednosti svim promenljivim iz skupa označava se vektorom $\boldsymbol{v}$. Vrednost konkretne promenljive V_s označava se kao v_s. Indikatorska funkcija $I(v_s = v')$ označava da je vrednost promenljive V_s upravo v'. Ako je v_s

fiksirana vrednost, izraz $\sum_{\boldsymbol{v}/v_s}$ označava sumiranje po svim mogućim dodelama $\boldsymbol{v}$ koje promenljivoj V_s dodeljuju vrednost v_s.

Pretpostavimo da raspodela verovatnoća p koju želimo da modeliramo može da se predstavi kao proizvod faktora - funkcija čiji su domeni neki podskupovi skupa svih promenljivih. Neka takvih podskupova ima A, i označimo ih sa $V_a \subseteq V$. Faktori su realne funkcije $\Psi_a : V_a \rightarrow R^+$. Ako je $\boldsymbol{v}_a$ vektor dodeljenih vrednosti promeljivima iz V_a, vrednost $\Psi_a(\boldsymbol{v}_a)$ označava koliko su konkretne dodele međusobno kompatibilne.

Neusmereni grafički model je familija rapodela verovatnoća od kojih se svaka faktoriše prema nekoj kolekciji faktora [4]. Drugim rečima, za datu kolekciju podskupova $\{V_a\}_{a = 1..A}$, i date funkcije Ψ_a, to je familija svih raspodela koje se mogu zapisati kao

$$p(\boldsymbol{v}) = \frac{1}{Z} \cdot \prod_{a=1}^{A} \Psi_a(\boldsymbol{v}_a) \quad , \tag{2}$$

pri čemu je Z normalizacioni faktor

$$Z = \sum_{\boldsymbol{y}} \prod_{a=1}^{A} \Psi_a(\boldsymbol{v}_a) \quad . \tag{3}$$

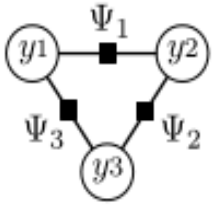
Faktorizacija oblika (2) može se lako predstaviti takozvanim faktor grafovima. Faktor graf je bipartitni graf G = (U, F, E), gde važi:

$U = \{1, 2, ..., m\}$ skup čvorova koji odgovaraju indeksima promenljivih u modelu
$F = \{1, 2, ..., A\}$ skup čvorova koji odgovaraju indeksima faktora u modelu
E = skup grana, pri čemu grana između čvora $s \in U$ i $a \in F$ označava da promenljiva $V_s \subseteq V_a$.

Raspodela verovatnoća $p(\boldsymbol{v})$ se faktoriše prema faktor grafu G [4] ako postoji skup lokalnih funkcija Ψ_a, tako da se p može zapisati kao

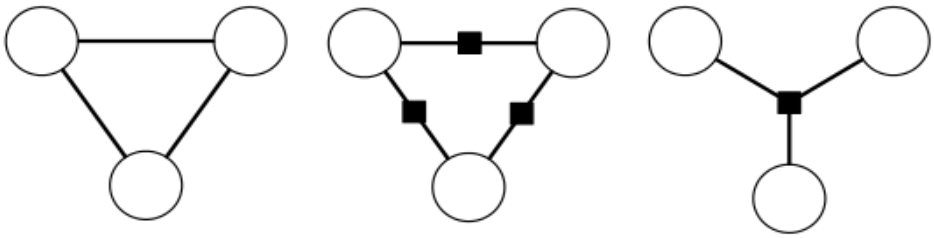
$$p(\boldsymbol{v}) = \frac{1}{Z} \cdot \prod_{a \in F} \Psi_a(\boldsymbol{v}_{N(a)}) \quad . \tag{4}$$

Grafički, čvorovi $s \in U$ se obično predstavljaju krugovima, dok se čvorovi $a \in F$ predstavljaju kvadratima. Jednostavan primer faktor grafa nad promenljivima y_1, y_2 i y_3, i faktorima Ψ_1, Ψ_2 i Ψ_3, prikazan je na slici 1. Model koji on definiše se faktoriše kao
$p(y_1, y_2, y_3) = \Psi_1(y_1, y_2)\Psi_2(y_2, y_3)\Psi_3(y_1, y_3)$.



Slika 1, primer faktor grafa, preuzeta iz [4]

Markovljeve mreže mogu prouzrokovati višeznačnu faktorizaciju, što nije slučaj sa faktor grafovima. Ovaj odnos ilustovan je slikom 2. Markovljeva mreža (levo) pokazuje da su promenljive v_1, v_2 *i* v_3 u nekom obliku zavisne, ali nije jasno da li se rapodela verovatnoća faktoriše kao $p(v_1, v_2, v_3) \propto f(v_1, v_2, v_3)$ ili kao $p(v_1, v_2, v_3) \propto f(v_1, v_2)g(v_2, v_3)h(v_1, v_3)$. Više o faktor grafovima može se naći u [5].



Slika 2, ilustracija višeznačnosti Markovljevih mreža, preuzeta iz [4]

2.2 Diskriminativne metode

Logistička regresija je diskriminativni model koji uopštava linearnu regresiju, i na osnovu koje se konstruišu drugi diskriminativni probabilistički modeli, poput modela maksimalne entropije, Markovljevih polja maksimalne entropije i uslovnih slučajnih polja. Ono što je zajedničko svim ovim modelima je što pri predikciji koriste određene parametre. Za različite vrednosti ovih parametara dobija se drugačija raspodela verovatnoća modela, te je potrebno pronaći najoptimalnije. Ovaj proces se naziva obučavanje parametara i jedna je od centralnih tema vezanih za mašinsko učenje. Kriterijum da li su određeni parametri dobri ili nisu daje skup za obučavanje. U diskriminativnom pristupu, dobri parametri su oni, koji za vrednosti sekvenci $\boldsymbol{x}^{(i)}$ iz skupa za obučavanje $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})$ predviđaju tagove $\boldsymbol{y}^{(i)\prime}$ najsličnije $\boldsymbol{y}^{(i)}$. Notacija u ovom poglavlju je preuzeta iz [7].

2.2.1 Linearna regresija

Pretpostavimo da imamo realne vektore $\boldsymbol{x}$ i $\boldsymbol{y}$, pri čemu je $\boldsymbol{x}$ vektor opservacija a $\boldsymbol{y}$ vektor nastao kao ishod tih opservacija. Zadatak je predstaviti $\boldsymbol{y}$ kao rezultat neke linearne funkcije opservacija $\boldsymbol{x}$. Dakle, potrebno je odrediti linearnu funkciju

$$y = w_0 + w_1 x \tag{5}$$

koja najbolje aproksimira unapred date vrednosti (x_i, y_i). Rešenje problema leži u pronalaženju parametara w_0 i w_1 koji najviše odgovaraju datim vrednostima. Jednom kada znamo ove parametre, a koji se mogu odrediti metodom najmanjih kvadrata [6], linearnu funkciju možemo primeniti na proizvoljne vrednosti $\boldsymbol{x}$, i time predvideti odgovarajuće vrednosti $\boldsymbol{y}$. Model koji rešava ovakve probleme je linearna regresija. On se može još uopštiti - $\boldsymbol{y}$ može zavisiti od nekoliko promenljivih $x^{(j)}$, od kojih je svaka realni vektor opservacija neke karakterike. Neka je broj takvih vektora K. U tom slučaju, potrebno je odrediti parametre w_j tako da vrednosi

$$y_i = w_0 + \sum_{j=0}^{K} w_j x_i^{(j)} \tag{6}$$

najbolje oslikavaju vrednosti $\boldsymbol{y}$ iz skupa za obučavanje. Problem (6) se ponovo može rešiti metodom najmanjih kvadrata. Primer ovakvog problema bi bila predikcija mesečnih primanja osobe, ako ona zavisi od radnog staža, trajanja školovanja i pola ($k=3$), i ukoliko za N osoba znamo ove karakteristike i njihova primanja.

2.2.2 Logistička regresija

Jedan od čestih problema koji se javljaju u mašinskom učenju je problem binarne klasifikacije. Želimo da razvijemo model koji će na osnovu nekih karakteristika podatka dati odgovor *da* ili *ne* na određeno pitanje, i pritom odrediti verovatnoću sa kojom daje odgovor. Ako je y binarna promenljiva koja zavisi od vektora realnih brojeva $\boldsymbol{x}$, želimo da odredimo $p(y=1|\boldsymbol{x})$. Ovakav model može se izgraditi iz linearne regresije, ali ne direktno. Ako je p broj između 0 i 1, funkcija definisana kao

$$logit(p) = \log \frac{p}{1-p} \tag{7}$$

naziva se logit funkcija. Logistička regresija je probabilistički model kojim se logit verovatnoća *p(y=1|**x**)* određuje modelom linearne regresije, odnosno

$$\log \frac{p(y=1|\boldsymbol{x})}{1-p(y=1|\boldsymbol{x})} = w_0 + \sum_{i=1}^{N} w_i x_i \quad . \tag{8}$$

Obe strane jednakosti (8) su realni brojevi, što opravdava okavav odabir modela. Odavde se zaključuje da je

$$\begin{aligned} p(y=1|\boldsymbol{x};w) &= \frac{1}{1+\exp\left(-w_0-\sum_{i=1}^{N} w_i x_i\right)} \\ p(y=0|\boldsymbol{x};w) &= \frac{\exp\left(-w_0-\sum_{i=1}^{N} w_i x_i\right)}{1+\exp\left(-w_0-\sum_{i=1}^{N} w_i x_i\right)} \end{aligned} \quad . \tag{9}$$

U navedenim izrazima, argument w označava da se procenjuje verovatnoća u odnosu na te fiksirane parametre. Najpogodniji parametri modela određuju se metodom maksimalne uslovne verodostojnosti, o čemu će biti reči u poglavlju 2.2.4.

2.2.3 Klasifikacija pomoću logističke regresije

Ako je poznat vektor karakteristika $\boldsymbol{x}$, za koji želimo da odredimo kojoj od dve klase verovatnije pripada, odnosno da li slučajna promeljiva y verovatnije uzima vrednost 1 ili 0, dovoljno je posmatrati linearnu funkciju po $\boldsymbol{x}$. Jasno je da bi promenljivoj y trebalo dodeliti vrednost 1 ako je

$$p(y=1|x,w) > p(y=0|x,w) \quad , \tag{10}$$

gde ćemo zanemariti slučaj da su ove dve verovatnoće jednake. Odavde sledi da treba dati potvrdan odgovor ako važe sledeće ekvivalentne jednakosti, iz kojih se jasnije vidi veza između linearne i logističke regresije:

$$\frac{p(y=1|x,w)}{p(y=0|x,w)} > 1 \tag{11}$$

$$\frac{p(y=1|x,w)}{1-p(y=1|x,w)}>1 \tag{12}$$

$$logit(p(y=1|x,w))>1 \tag{13}$$

$$\exp(w_0+\sum_{i=1}^{N} w_i x_i)>1 \tag{14}$$

$$w_0+\sum_{i=1}^{N} w_i x_i>0 \quad . \tag{15}$$

Dakle, jednom kada su parametri *w* poznati, lako je dati traženi odgovor. U nastavku rada, smatraće se da x_0 ima vrednost 1, radi kompaktnijeg zapisa.

2.2.4 Procena parametara i princip maksimalne verodostojnosti

U svakom probabilističkom modelu učestvuje određen broj parametara, na osnovu kojih se vrši predikcija. Oni nisu unapred poznati, i potrebno je odrediti ih na osnovu skupa za obučavanje. Uslovna slučajna polja nasleđuju metodu za obučavanje parametara od logističke regresije, takozvani princip maksimalne verodostojnosti.

Neka je data familija raspodela verovatnoća definisana vektorom parametara Θ, i neka je dat slučajan uzorak nezavisnih ishoda $(x^{(1)}, x^{(2)}, ..., x^{(n)})$ jedne od tih raspodela. Cilj je odrediti konkretnu raspodelu kojoj taj uzorak pripada, odnosno odrediti Θ. Raspodela verovatnoća je definisana funkcijom ili gustinom raspodele, u zavisnosti od tipa promenljivih. Neka je f_Θ verovatnoća definisana nekom od tih funkcija, za svaku moguću vrednost Θ. Kako su ishodi $x^{(i)}$ međusobno nezavisni, za fiksiranu vrednost Θ, verovatnoća celog uzorka se računa kao

$$f(x^{(1)},...,x^{(n)};\Theta)=\prod_j f_\Theta(x^{(j)},\Theta) \quad . \tag{16}$$

Ako, sa druge strane, posmatramo uzorak kao fiksan, za različite vrednosti paramtera Θ dobijaju se različite vrednosti izraza (16). Funkcija verodostojnosti zavisi od tih parametara i definiše se kao [8]

$$L(\Theta; x^{(1)},...,x^{(n)})=f(x^{(1)},...,x^{(n)};\Theta) \quad . \tag{17}$$

Prema principu maksimalne verodostojnosti, parametre Θ treba odabrati tako da funkcija

verodostojnosti bude što veća, odnosno traženi parametri treba da zadovoljavaju

$$\Theta^* = argmax_\Theta L(\Theta; x_1, \ldots, x_n) \quad . \tag{18}$$

U slučaju logističke regresije, verovatnoće koje se posmatraju su uslovne, te je ovaj princip potrebno malo promeniti. Funkcija uslovne verodostojnosti parametara, za date slučajne promenljive x i y definiše se kao

$$L(\Theta; y|x) = f(y|x;\Theta) \quad . \tag{19}$$

U ovom slučaju, raspodela slučajne promenljive x nije od značaja, kao ni zahtev da su ishodi $x^{(i)}$ međusobno nezavisni. Potrebno je, međutim, da ishodi $y^{(i)}$ budu međusobno nezavisni, uslovljeni vrednostima $x^{(i)}$. Kada je poznat skup sekvenci za obučavanje $\{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\}_{i=1,\ldots,N}$, odgovarajući parametri se traže tako da maksimizuju $\prod_i f(y^{(i)}|x^{(i)};\Theta)$. Ponekad je poželjno posmatrati logaritamsku uslovnu verodostojnost, definisanu

$$l(\Theta) = \sum_{i=1}^{N} \log f(y^{(i)}|x^{(i)};\Theta) \quad , \tag{20}$$

pri čemu je cilj proceniti parametre θ^* tako da

$$\Theta^* = argmax_\Theta l(\Theta) \quad . \tag{21}$$

Same formule za obučavanje parametara logističke regresije ovde neće biti navedene, ali će se njihovim varijantama za uslovna slučajna polja baviti poglavlje 2.8.1.

2.2.5 Model maksimalne entropije

Varijanta linearne regresije u kojoj se određuje pripadnost sekvence jednoj od klasa $C = \{c_1, c_2, \ldots c_k\}$ naziva se multinomialna logistička regresija, ili model maksimalne entropije.
Neka je y odgovarajuća klasa sekvence $\boldsymbol{x}$. Prema modelu maksimalne entropije [7], verovatnoća da y uzme neku konkretnu vrednost $c \in C$ je

$$p(c|x) = \frac{1}{Z} \exp\left(\sum_{i=0}^{K} w_{ci} f_i(x, c)\right) \quad , \tag{22}$$

gde je Z faktor normalizacije koji obezbeđuje da je izrazom (22) korektno definisina

verovatnoća, odnosno da je suma verovatnoća svih klasa za datu sekvencu jednaka 1:

$$Z = \sum_{c' \in C} \exp\left(\sum_{i=0}^{K} w_{c'i} f_i(x, c)\right) \quad . \tag{23}$$

Smatra se da svaka od klasa c ima svoje parametre. Funkcije f_i su indikatorske funkcije koje oslikavaju neko svojstvo date sekvence i klase, i nazivaju se i karakteristične funkcije. Ako su poznati parametri modela, i dat skup opservacija x, zadatak je naći klasu kojoj ona najverovatnije pripada, odnosno

$$\hat{c} = argmax_{c \in C}\, p(c|x) \quad . \tag{24}$$

Ovaj postupak podrazumeva izgradnju posebnog linearnog modela za svaku od klasa [7]. Učenje parametara ovog modela obavlja se na sličan način kao u slučaju logističke regresije, tako da na skupu za obučavanje veličine N traže

$$\hat{w} = argmax_{w} \prod_{i=1}^{N} p\left(y^{(i)}|x^{(i)}\right) \quad . \tag{25}$$

Jedan od čestih problema probabilističkih modela u kojima učestvuje veliki broj parametara je overfitting, koji nastaje kada se model previše oslanja na skup za obučavanje, i gubi mogućnost precizne predikcije. Uobičajeni način prevazilaženja overfittinga je regularizacija, kojom se smanjivanjem vrednosti parametara redukuje i fleksibilnost celokupnog modela

$$\hat{w} = argmax_{w} \prod_{i=1}^{M} p\left(y^{(i)}|x^{(i)}\right) - \alpha \cdot R(w) \quad , \tag{26}$$

gde R uglavnom neki oblik norme vektora w.

Označimo sa P prostor dopustivih raspodela verovatnoća koje modeliraju skup za obučavanje, ili u ovom konkretnom slučaju zavisnost promenljivih y od x na osnovu skupa za obučavanje. Zadatak predikcije je odrediti najverniju od tih verovatnoća. Svaka od definisanih karakterističnih funkcija smanjuje prostor P zahtevajući da njena vrednost u razvijenom modelu oponaša vrednosti na skupu za obučavanje. Princip maksimalne entropije nalaže da je tražena raspodela verovatnoća upravo ona raspodela iz P čija je entropija najveća, odnosno najunifomnija raspodela iz P. Ovo znači da model ne bi trebalo da ima bilo kakve pretpostavke o podacima osim onih podrazumevanih karakterističnim funkcijama. Drugim rečima, ne treba pretpostavljati ništa o podacima van skupa za obučavanje. Raspodela verovatnoća definisana izrazom (22), za optimalne vrednosti parametara, je

upravo najuniformnija raspodela koja poštuje uslove definisane karakterističnim funkcijama. Više o modelu maksimalne entropije može se naći u [9].

2.3 Skriveni Markovljevi modeli

Markovljevi lanci su jednostavan probabilistički model koji se može definisati kao težinski automat sa skupom stanja S kojim se generiše sekvenca stanja $\boldsymbol{y} = (y_1, y_2, ..., y_T)$. Za svaka dva stanja $y', y'' \in Q$, verovatnoća prelaska iz y'' u y' definiše se kao

$$p(y_i = y' | y_{i-1} = y'') \quad . \tag{27}$$

Izraz (27) se kraće zapisuje i kao

$$p(y' | y'') \quad . \tag{28}$$

Pretpostavka modela je da je prelazak iz jednog stanja u sledeće uslovljeno jedino trenutnim stanjem.

$$p(y_{i+1} | y_1, ..., y_i) = p(y_{i+1} | y_i) \quad . \tag{29}$$

Verovatnoća generisanja konkretne sekvence $\boldsymbol{y} = (y_1, y_2, ..., y_T)$ predstavlja se kao

$$\begin{aligned} p(\boldsymbol{y}) &= p(y_T, y_{T-1}, ..., y_1) = p(y_T | y_{T-1}, ..., y_1) \cdot p(y_{T-1} | y_{T-2}, ..., y_1) \cdot ... \cdot p(y_1) \\ &= p(y_T | y_{T-1}) \cdot p(y_{T-1} | y_{T-2}) \cdot ... \cdot p(y_2 | y_1) \cdot p(y_1) \\ &= \prod_{t=1}^{T} p(y_t | y_{t-1}) \end{aligned} \quad . \tag{30}$$

Ova notacija podrazumeva uvođenje tihog stanja y_0, na osnovu kojih se definiše početna raspodela verovatnoća stanja $p(y_k | y_0)$. Početne verovatnoće i verovatnoće prelaza su zapravo parametri modela. Primena Markovljevih lanaca u bioinformatici može se naći u [10].

Markovljevi lanci su korisni ukoliko su poznata sva stanja koja su generisala datu sekvencu, što nije uvek slučaj. Često je poznata samo opservacija $\boldsymbol{x}$, koja je mogla biti generisana na više načina, a sama putanja stanja je skrivena. Ovakvim problemima bave se skriveni Markovljevi modeli. Iz dosadašnje priče jasno je da se radi o generativnom prababilističkom modelu. Nastao je decenijama pre CRF, dok sami Markovljevi lanci datiraju još iz 1906. [11]

Kod skrivenih Markovljevih modela razlikuju se dve sekvence: jedna sekvenca je opservacija, odnosno sekvenca generisanih simbola, koja je poznata, i koju ćemo označiti kao $x = (x_1, ..., x_T)$. Druga je sekvenca stanja koja su zapravo generisala date simbole, i ona je skrivena. Označimo je sa $y = (y_1, ..., y_T)$. Neka je S konačan skup dopustivih stanja, a O

konačan skup mogućih opservacija. Sama sekvenca stanja je Markovljev lanac, tako da je prelaz iz stanja koje je generisalo simbol x_{t-1} u stanje koje generiše x_t uslovljen samo trenutnim stanjem. Verovatnoće prelaza za konkretne vrednosti y_t *i* y_{t-1} definisane su kao

$$p(y_t|y_{t-1}) \quad . \tag{31}$$

Podrazumevaju se prethodno definisane početne verovatnoće. Verovatnoća da je simbol x_t generisan u stanju y_t je emisiona verovatnoća

$$p(x_t|y_t) \quad . \tag{32}$$

Za date sekvence x i y lako je odrediti zajedničku verovatnoću [4]

$$p(x,y)=\prod_{t=1}^{T} p(y_t|y_{t-1})\,p(x_t|y_t) \quad . \tag{33}$$

Za skrivene Markovljeve modele vezuju se tri problema:

1) Učenje parametara - na osnovu datog skupa za obučavanje odrediti parametre modela, odnosno verovanoće $p(x_t|y_t)$ *i* $p(y_t|y_{t-1})$. U ove svrhe koristi se Baum-Welchov algoritam [10].
2) Za datu sekvencu x odrediti najverovatniju putanju $\boldsymbol{y}$. Ovaj problem bi mogao da se reši tako sto bi se računala zajednička verovatnoća $\boldsymbol{x}$ i svake moguće putanje, ali je ovaj pristup neefikasan jer broj ovakvih putanja raste eksponencijalno sa dužinom sekvence $\boldsymbol{x}$. U praksi, najverovatnija putanja određuje se algoritmom dinamičkog programiranja - Viterbi algoritmom.
3) Poslednji problem je pronalaženje verovatnoće pojavljivanja same niske $\boldsymbol{x}$. Ona bi mogla da se dobije sumiranjem zajedničkih verovatnoća $\boldsymbol{x}$ i svih mogućih putanja, što opet nije efikasno. Za rešavanje ovog problema koriste se algoritmi dinamičkog programiranja unapred i unazad.

Skriveni Markovljev model se može videti i kao faktor graf kojim se definiše raspodela verovatnoća [4]

$$p(x,y)=\prod_{t} \Psi_t(x_t, y_t, y_{t-1}) \quad , \tag{34}$$

gde je

$$Z=1 \quad , \tag{35}$$

$$\Psi_t(x, j, i) = p(y_t = j | y_{t-1} = i)\, p(x_t = x | y_t = j) \quad . \tag{36}$$

Verovatnoća sekvence $\boldsymbol{x}$ računa se algoritmom unapred. Ona se može posmatrati kao

$$\begin{aligned} p(\boldsymbol{x}) &= \sum_{y} p(\boldsymbol{x}, y) = \sum_{\boldsymbol{y}} \prod_{t=1}^{T} \Psi_t(x_t, y_t, y_{t-1}) \\ &= \sum_{yT} \sum_{yT-1} \Psi_t(x_T, y_T, y_{T-1}) \sum_{yT-2} \Psi_t(x_{t-1}, y_{t-1}, y_{t-2}) \sum_{yT-3} \ldots \end{aligned} \quad . \tag{37}$$

Kako se veliki broj sabiraka u izrazu (37) ponavlja veći broj puta, algoritam dinamičkog programiranja kojim se ova verovatnoća računa bi bio adekvatan. Definišimo skup promenljivih $\alpha_t\ (j)$ kao verovatnoću da je generisano t simbola sekvence $\boldsymbol{x}$, pri čemu je poslednji simbol generisan u stanju j

$$\alpha_t(j) = p(x_{1,} x_{2,} \ldots, x_t, y_t = j) \quad , \tag{38}$$

$$\alpha_t(j) = \sum_{y_{1,} \cdots, y_{t-1}} \Psi_t(x_t, j, y_{t-1}) \prod_{t'=1}^{t-1} \Psi_{t'}(x_{t'}, y_{t'}, y_{t'-1}) \quad , \tag{39}$$

odnosno, rekurzivno izraženo

$$\begin{aligned} \alpha_t(j) &= \sum_{y_{1,} \cdots, y_{t-1}} \Psi_t(x_t, j, i) \alpha_{t-1}(i) \\ \alpha_1(j) &= \Psi_1(x_{1,} j, y_0) \end{aligned} \quad . \tag{40}$$

Kako je verovatnoća sekvence $\boldsymbol{x}$ zapravo verovatnoća da pročitamo celu sekvencu i dođemo u završno stanje, po bilo kojoj sekvenci stanja, potrebna nam je vrednost

$$p(x) = \sum_{y_T} \alpha_T(y_T) \quad . \tag{41}$$

Slično ovome, definišimo promenljive $\beta_t\ (i)$ kao verovatnoću da se generiše ostatak sekvence $\boldsymbol{x}$ ako je trenutno stanje i

$$\beta_t(i) = p(x_{t+1}, x_{t+2}, \ldots, x_T | y_t = i) \quad , \tag{42}$$

$$\beta_t(i)=\sum_{y_{t+1,\dots,T}}\prod_{t'=t+1}^{T}\Psi_{t'}(x_{t'},y_{t'},y_{t'-1}) \quad , \tag{43}$$

osnosno

$$\begin{matrix}\beta_t(i)=\sum j\in S\,\Psi_{t+1}(x_{t+1},j,i)\beta_{t+1}(j)\\ \beta_T(i)=1\end{matrix} \quad . \tag{44}$$

Analogno, verovanoća sekvence se može izračunati kao

$$p(x)=\beta_0(y_0)=\sum_{y1}\Psi_1(x_{1,}y_{1,}y_0)\beta_1(y_1) \tag{45}$$

Ove vrednosti se koriste pri računanju marginalnih verovatnoća $p(y_{t-1}, y_t|\boldsymbol{x})$ na sledeći način:

$$\begin{aligned}p(y_{t-1},y_t|x)&=\frac{1}{p(\boldsymbol{x})}p(\boldsymbol{x},y_{t-1},y_t)=\frac{1}{p(\boldsymbol{x})}p(x_{1,}\dots,x_{t-1},x_t,x_{t+1},y_{t-1},y_t)\\ &=\frac{1}{p(\boldsymbol{x})}p(x_{1,}\dots,x_{t-1},y_{t-1},y_t)p(x_t,x_{t+1},\dots,x_T|x_{1,}..,x_{t-1},y_{t-1},y_t)\\ =\frac{1}{p(\boldsymbol{x})}p(x_{1,}\dots,x_{t-1},y_{t-1})p(y_t|x_{1,}\dots,x_{t-1},y_{t-1})p(x_t|x_{1,}\dots,x_{t-1},y_{t-1},y_t)p(x_{t+1},\dots,x_T|&\\ &=\frac{1}{p(\boldsymbol{x})}p(x_{1,}\dots,x_{t-1},y_{t-1})p(y_t|y_{t-1})p(x_t|y_t)p(x_{t+1},\dots,x_T|y_t)\\ &=\frac{1}{p(\boldsymbol{x})}\alpha_{t-1}(y_{t-1})\Psi_t(x_t,y_t,y_{t-1})\beta_t(y_t)\end{aligned} \tag{46}$$

Algoritam unapred-unazad radi tako što prvo izračuna vrednosti $\alpha_t\,(j)$ i $\beta_t\,(i)$, i nakon toga, marginalne verovatnoće definisane izrazom (45). Računanje vrednosti $\alpha_t\,(j)$ je zapravo algoritam unapred, dok je računanje $\beta_t\,(i)$ algoritam unazad.

Najverovatnija sekvenca stanja se može pronaći algoritmom sličnim algortimu unapred, zamenjivanjem sumiranja maksimizacijom:

$$\delta_t(j)=max_{y_{1,}\dots,y_{t-1}}\Psi_t(x_t,j,y_{t-1})\prod_{t'=1}^{t-1}\Psi_{t'}(x_{t'},y_{t'},y_{t'-1}) \quad , \tag{47}$$

ili rekurzivno

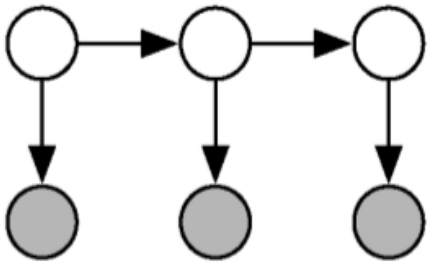
$$\delta_t(j)=max_{i\in S}\Psi_t(x_t,j,i)\delta_{t-1}(i) \quad . \tag{48}$$

Kada se vrednosti $\beta_t\,(i)$ izračunaju, najverovatnija sekvenca stanja se može prinaći kao

$$\begin{aligned} y_t^* &= argmax_{i \in S} \delta_T(i) \\ y_t^* &= argmax_{i \in S} \Psi_t(x_{t+1}, y_{t+1}^*, i) \delta_t(i), \quad t<T \end{aligned} \quad . \tag{49}$$

Rekurzivne formule za računanje β_t *(i)* i y_t^* su zapravo Viterbi algoritam. Složenost predstavljenih algoritama dinamičkog programiranja je $O(TM^2)$, gde je T dužina sekvence a M broj mogućih tagova.

Jednostavan skriven Markovljev model predstavljen je slikom 3. Beli čvorovi označavaju promenljive y dok su suvi čvorovi promenljive *x*.Postoje dva skupa usmerenih grana: grana od stanja y_{t-1} ka narednom stanju y_t označava da je svako stanje uslovljeno prethodnim, dok grana od y_t ka x_t oslikava da je simbol uslovljen stanjem u kojem je generisan.



Slika 3, skriveni Markovljev model, preuzeta iz [4]

2.4 Markovljevi modeli maksimalne entropije

Model maksimalne entropije obezbedio je mehanizam za klasifikaciju podataka, dok se skriveni Markovljevi modeli bave sekvencijalnom predikcijom. Ideja iza Markovljevih modela maksimalne entropije, kao i uslovnih slučajnih polja, je kombinovanje ova dva pristupa, čime bi se omogućila poboljšana sekvencijalna klasifikacija podataka. Markovljevi lanci, koji su u osnovi skrivenih Markovljevih modela, umogome ograničavaju izbor karakteristika. Tačnije, jedine karakteristike koje je dozvoljeno koristiti su prethodno stanje (tag) i trenutni simbol (reč), na osnovu kojih se određuje sledeće stanje u predikciji. U ovakav model je teško uključiti karakteristike poput prefiksa reči, koje su jako korisne u procesiranju prirodnih jezika. Sa druge strane, to je lako izvodljivo uvođenjem indikatorske funkcije koja bi odražavala neko svojsvo konkretnog prefiksa. Markovljev model maksimalne entropije je dikriminativna metoda zasnovana na skrivenim Markovljevim modelima, pri čemu se prelazak iz jednog stanja u drugo ne računa preko emisionih i verovatnoća prelaza, već modelom maksimalne entropije, koja se za određene zadatke pokazala kao dosta efikasnija od svog prethodnika [12].

Formalnije, ako je $\boldsymbol{x} = (x_1, x_2, ..., x_T)$ sekvenca opservacija, Markovljev model maskimalne entropije u najopštijem slučaju definiše verovatnoću konkretne sekvence stanja $\boldsymbol{y} = (y_1, ..., y_T)$ kao

$$p(y|x)=\prod_{t=1}^{T} p(y_t|y_{t-1}, x_t) \quad , \tag{50}$$

gde se, za unapred definisane funkcije f_k, verovatnoća prelaza definiše se modelom maksimalne entropije

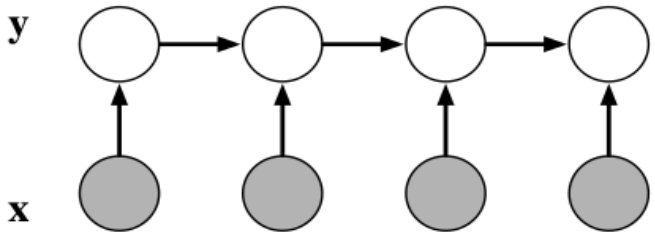
$$p(y_t|y_{t-1}, x)=\frac{1}{Z_t(x, y_{t-1})}\exp\sum_k w_k f_k(x, y_t, y_{t-1}) \quad , \tag{51}$$

$$Z_t(x, y_{t-1})=\sum_{y'} \exp[\sum_k w_k f_k(x, y', y_{t-1})] \quad . \tag{52}$$

Za razliku od skrivenih Markovljevih modela (33), gde su verovatnoće prelaska i emisija bile razdvojene, sada svako stanje zavisi od prethodnog stanja i trenutnog simbola posmatranih zajedno. Pri određivanju najverovatnije putanje $\boldsymbol{y}$ koristi se modifikovana verzija Viterbi algoritma, dok se obučavanje parametara odvija na sličan način kao kod diskriminativnih modela logističke regresije i maksimalne entropije.

U nekim situacijama se dešava da pojedina stanja imaju mali broj izlazećih grana, pa tranzicione verovatnoće tih stanja imaju malu entropiju. Drugim rečima, kako je suma svih verovatnoća prelaska iz jednog stanja 1, ako je mali broj tih verovatnoća pozitivan, te verovatnoće su jako velike. U najgorem slučaju, postoji samo jedna izlazna grana. Kada se ovo dogodi, same opservaije igraju malu ili nikakvu ulogu pri prelasku u novo stanje. Problem leži u činjenici da se model maskimalne entropije konstruiše lokalno, za svaki prelaz pojedinačno. Ovaj problem javlja se kod Markovljevih polja maksimalne entropije, i zove se problem naklonosti tagovima [2]. Uslovna slučajna polja nastala su baš kako bi se on prevazišao, zadržavajući ostale prednosti Markovljevih modela maksimalne entropije nad skrivenim Markovljevim modelima [1].

Primer Markovljevog modela maksimalne entropije dat je na slici 4. Za razliku od slike 3, grane su usmerene od x_t ka y_t, jer sada opservacije uslovljavaju stanja.



Slika 4, Markovljev model maksimalne entropije, preuzeta iz [4]

2.5 Linearna uslovna slučajna polja

Neka je S skup svih mogućih tagova, *O* skup opservacija, a indikatorska funkcija $I(y_t = y')$ uzima vrednost 1 ako slučajna promenljiva y_t ima vrednost y', i 0 inače. Izraz (33), koji definiše raspodelu verovatnoća skrivenih Markovljevih modela, se može generalizovati kao [4]

$$p(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{Z} \prod_{t=1}^{L} \exp[\sum_{y', y'' \in S} w_{y'y''} \cdot I(y_t = y') \cdot I(y_{t-1} = y'') + \sum_{y' \in S} \sum_{x' \in O} \mu_{x'y'} \cdot I(y_t = y') \cdot I(x_t = \quad (53)$$

gde su $\Theta = \{w_{ij}, \mu_{oi}\}$ parametri modela, a Z je konstanta normalizacije koja obezbeđuje da izraz (51) definiše verovatnoću. Izrazi (53) i (33) definišu istu raspodelu verovatnoća ukoliko važe sledeće jednakosti

$$\begin{gathered} w_{y'y''} = \log p(y_t = y' | y_{t-1} = y'') \\ \mu_{x'y'} = \log p(x_t = x' | y_t = y') \\ Z = 1 \end{gathered} \quad . \quad (54)$$

Jedan od najbitnijih pojmova linearnih uslovnih slučanih polja su karakteristične funkcije - indikatorske funkcije koje zavise od dva uzastopna taga i nekih karakteristika sekvence $\boldsymbol{x}$. Za početak, definišimo ih kao

$$f_{y'y''}(y_t, t_{t-1}, x_t) = I(y_t = y') \cdot I(y_{t-1} = y'') \quad (55)$$

i

$$f_{x'y'}(y_t, t_{t-1}, x_t) = I(y_t = y') \cdot I(x_t = x') \quad , \quad (56)$$

za svaki par tagova *(y', y'')*, odnosno par opservacije i taga *(x', y')*. Karakterisične funkcije

oblika $f_k(y_t, y_{t-1}, x_t)$ zapravo su proizvodi prethodno definisanih funkcija za sve vrednosti x', y' i y''. Izraz (53) se sada može zapisati kao

$$p(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{Z} \prod_{t=1}^{T} \exp[\sum_{k=1}^{K} w_k f_k(y_t, y_{t-1}, x_t)] \tag{57}$$

Izraz (57) i dalje definiše istu raspodelu verovatnoća kao standardni skriveni Markovljevi modeli, pri uslovima (54). Kako su linearna uslovna polja kojim težimo diskriminativni model, ostalo je još samo da umesto zajedničke posmatramo uslovnu verovatnoću

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}, \boldsymbol{y})}{\sum_{y'} p(\boldsymbol{x}, \boldsymbol{y}')} = \frac{\prod_{t=1}^{T} \exp[\sum_{k=1}^{K} w_k f_k(y_t, y_{t-1}, x_t)]}{\sum_{y'} \prod_{t=1}^{T} \exp[\sum_{k=1}^{K} w_k f_k(y'_t, y'_{t-1}, x_t)]} \tag{58}$$

Uslovna raspodela definisana izrazom (58) je specijalan slučaj linearnih slučajnih polja, gde karakteristične funkcije posmatraju samo identitete karateristika $\boldsymbol{x}$ i tagova $\boldsymbol{y}$. Prednost uslovnih slučanih polja nad skrivenim Markovljevim modelima je što se ove funkcije mogu lako uopštiti tako da odražavaju bilo koje svojstvo karakteristika.

Neka su X i Y slučajne promenljive, $w = \{w_k\}$ realni vektor parametara, i $F = \{ f_k(y_t, y_{t-1}, \boldsymbol{x}_t)\}_{k=1}^{K}$ skup realnih karakterističnih funkcija. Linearna uslovna slučajna polja definišu se [4] uslovnom raspodelom

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(x)} \prod_{t=1}^{T} \exp[\sum_{k=1}^{K} w_k f_k(y_t, y_{t-1}, \boldsymbol{x}_t)] \quad , \tag{59}$$

pri čemu je

$$Z(\boldsymbol{x}) = \sum_{y} \prod_{t=1}^{T} \exp[\sum_{k=1}^{K} w_k f_k(y_t, y_{t-1}, \boldsymbol{x}_t)] \quad . \tag{60}$$

Karakteristična funkcija može posmatrati proizvoljne karakteristike sekvence, tako da se vektor x_t zapravo može zameniti celokupnom sekvencom opservacija $\boldsymbol{x}$.

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(x)} \prod_{t=1}^{T} \exp[\sum_{k=1}^{K} w_k f_k(y_t, y_{t-1}, \boldsymbol{x})] \quad , \tag{61}$$

$$Z(\boldsymbol{x}) = \sum_{y} \prod_{t=1}^{T} \exp[\sum_{k=1}^{K} w_k f_k(y_t, y_{t-1}, \boldsymbol{x})] \quad . \tag{62}$$

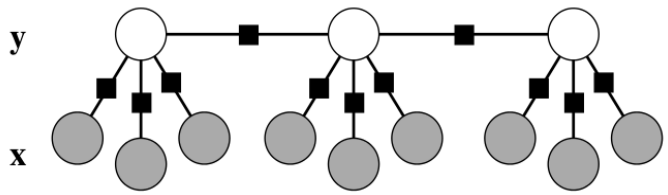
Linearna uslovna slučajna polja omogućavaju posmatranje samo dva uzastopna taga, zbog čega se model može posmatrati kao lanac. Ovo ograničenje prevazilazi se kod generalnih uslovnih slučajnih polja.

Sa druge strane, linearna uslovna slučajna polja mogu se definisati i preko faktor grafa (2) nad skupom čvorova $U = X \cup Y$ kao

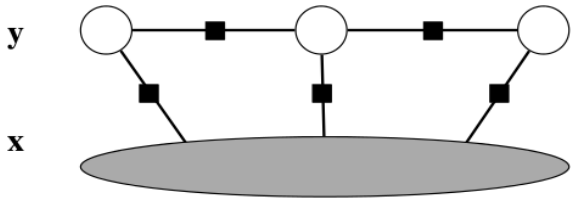
$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z}(\boldsymbol{x}) \cdot \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, \boldsymbol{x}_t) \quad , \tag{63}$$

$$\Psi_t(y_t, y_{t-1}, \boldsymbol{x}_t) = \exp\left[\sum_{k=1}^{K} w_k f_k(\boldsymbol{x}_t, y_t, y_{t-1})\right] \quad . \tag{64}$$

Pojednostavljena linearna uslovna slučajna polja predstavljena izrazom (58) ilustrovana su slikom 5. U praksi, moguće je posmatranje proizvoljnih karakteristika, pa bi adekvatan faktor graf za raspodelu verovatnoća (61) mogao da se predstavi slikom 6.



Slika 5, linearlna uslovna slučajna polja, preuzeta iz [4]



Slika 6, linearlna uslovna slučajna polja, preuzeta iz [4]

Iz izraza (61) i (62) vidi se da su linearna uslovna slučajna polja usko vezana za modele maksimalne entropije i logističke regresije. Takođe, vidljiva je povezanost sa Markovljevim modelima maksimalne entropije (51). Razlika između ova dva diskriminativna modela je u tome što se kod Markovljenih modela maksimalne entropije za svako stanje konstruiše pojedinačan eksponencijalni model za verovatnoće prelaza u naredno stanje, dok kod linearnih uslovnih slučajnih polja jedan eksponencijalni model određuje verovatnoće cele sekvence tagova [1]. Analogno tome, svaki od ovih eksponencijalnih modela je normalizovan, ali se kod Markovljevih modela maksimalne entropije radi o normalizaciji verovatnoća prelaza iz svakog pojedinčanog stanja (52), dok se u slučaju uslovnih slučajnih polja normalizacija vrši za celu sekvencu tagova (62). Efikasnost i preciznost metoda uslovnih slučajnih polja, skrivenih Markovljevih modela i Markovljevih modela maskimalne entropije upoređeni su u radovima [1][13], u kojima je prednost data dikriminativnim

metodama.

Funkcije f_k se nazivaju i indikatorske karakteristične funkcije niskog nivoa, čijim se sumiranjem dobijaju karakteristične funkcije oblika

$$F_k(\boldsymbol{x}, \boldsymbol{y}) = \sum_i f_k(y_i, y_{i-1}, \boldsymbol{x}) \tag{65}$$

koje posmatraju cele sekvence $\boldsymbol{x}$ i $\boldsymbol{y}$. Izraz () sada izgleda

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp[\sum_{k=1}^{K} w_k F_k(\boldsymbol{x}, \boldsymbol{y})] \quad , \tag{66}$$

$$Z(\boldsymbol{x}) = \sum_{y'} \exp[\sum_{k=1}^{K} w_k F_k(\boldsymbol{x}, \boldsymbol{y}')] \quad . \tag{67}$$

Odabir karakterističnih funkcija vezan je za konkretan problem koji se rešava. U primeru određivanja imenovanih entiteta, funkcija nižeg nivoa mogla bi da glasi "prethodna reč je "Novi", trenutna reč je "Sad", prethodni tag je "lokacija" i trenutni tag je "lokacija"".

Svakoj karakterističnoj funkciji odgovara jedan parametar, koji se određuje u procesu obučavanja. Pojednostavljeno, što je taj parametar veći, to ta funkcija bolje opisuje skup za obučavanje.

2.6 Generalna uslovna slučajna polja

Generalna uslovna slučajna polja lako se uopštavaju iz linearnih, korišćenjem opštijih faktor grafova.

Neka je *G* faktor graf nad promenljivima *X* i *Y*. *(X,Y)* je uslovno slučajno polje ako se za neke vrednosti $\boldsymbol{x}$ promenljivih *X*, uslovna verovatnoća $p(\boldsymbol{y}|\boldsymbol{x})$ faktoriše prema *G* [4].

Svako uslovno slučajno polje se, dakle, faktoriše po nekom faktor grafu *G*. Dati faktor graf *G*, i njegov skup faktora $\{\Psi_a\}$, definišu raspodelu

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z}(\boldsymbol{x}) \cdot \prod_{t=1}^{T} \Psi_a(x_a, y_a) \quad , \tag{68}$$

gde je

$$\Psi_a(x_a, y_a) = \exp[\sum_{k=1}^{K} w_{ak} f_{ak}(x_a, y_a)] \quad . \tag{69}$$

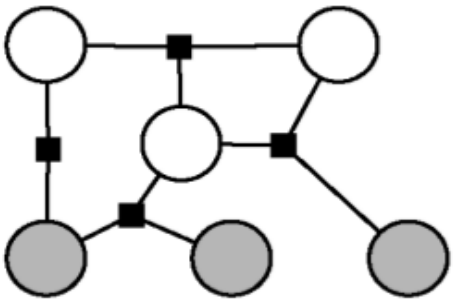
gde u indeksiranju karakterističnih funkcija i odgovarajućih parametara učestvuje i oznaka faktora *a*, jer svaki faktor može da ima svoj skup parametara - čak i karakterisitčnih funkcija. Faktor Ψ_a zavisi od unije nekih podskupova $X_a \subseteq X$ i $Y_a \subseteq Y$. U slučaju linearnih uslovnih slučajnih polja, skup Y_a je mogao da sadrži samo dva susedna taga.

Korisno je grupisati faktore koje koriste iste karakteristične funkcije i njima odgovarajuće parametre. Skup svih parametara može se podeliti na klase $C = \{C_1, C_2, ..., C_P\}$, gde svaka klasa C_P koristi isti skup karakterističnih funkcija $\{f_{pk}(x_c, y_c)\}_{k=1...K(p)}$, i vektor parametara w_{kp} , veličine $K(p)$. Svaka od ovih klasa se naziva klika šablona. Raspodela verovatnoća uslovnih slučajnih polja se sada može napisati kao

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z}(\boldsymbol{x}) \cdot \prod_{C_p \in C} \prod_{\Psi_c \in C_p} \Psi_c(x_c, y_c; w_p) \quad , \tag{70}$$

$$\Psi_c(x_c, y_c; w_p) = \exp[\sum_{k=1}^{K} (p) w_{pk} f_{pk}(x_c, y_c)] \quad . \tag{71}$$

Jednostavan model uslovnih slučajnih polja ilustrovan je slikom 7. Sivi čvorovi su promenljive *x*, beli čvorovi odgovaraju tagovima, dok su faktori predstavljeni crnim kvadratima.



Slika 7, generalna uslovna slučajna polja, preuzeta iz [4]

2.7 Algoritmi

2.7.1 Pregled algoritama

Za dati model (68), obučavanje parametara je proces odabira realnog vektora *w* koji daje najbolju moguću predikciju

$$\hat{\boldsymbol{y}} = argmax_{y}\, p(\boldsymbol{y}|\boldsymbol{x}; w) \tag{72}$$

na osnovu datog skupa za obučavanje $D = \{x^{(i)}, y^{(i)}\}_{i = 1, .., N}$. Poželjno je naći takve parametre *w*, da su vrednosti tagova dobijene predikcijom najsličnije onima datim u skupu za obučavanje. Radi ilustracije, posmatrajmo linearna uslovna slučajna polja i jednu konkretnu karakterističnu funkciju, f_k. Cilj je da vrednost ove funkcije na celom skupu za obučavanje $\{x^{(i)}, y^{(i)}\}$ bude jednaka njenoj vrednosti na skupu $\{x^{(i)}, y'\}$, gde su vrednosti y' dobijene predikcijom, odnosno

$$\sum_{i=1}^{N}\sum_{l=1}^{L} f_k(y_t^{(i)}, y_{t-1}^{(i)}, \boldsymbol{x}_t^{(i)}) = \sum_{i=1}^{N}\sum_{l=1}^{L}\sum_{y,y'} f_k(y, y', \boldsymbol{x}_t^{(i)}) \cdot p(y_t = y, y_{t-1} = y' | \boldsymbol{x}^{(i)}) \quad . \tag{73}$$

Ovaj problem se rešava metodom maksimalne verodostojnosti, ali u tu svrhu postoje i druge razvijene metode, o čemu će biti reči u nastavku. Međutim, učenje parametara zahteva efikasne algoritme za rešavanje problema zaključivanja (engl *inference*) koji se dele u dve grupe:

- Traženje najverovantijeg taga za datu sekvencu x.
- Računanje marginalnih verovatnoća oblika $p(y'|\boldsymbol{x})$ i $p(y',y''|\boldsymbol{x})$, čime se uglavnom izračunava i faktor normalizacije $Z(x)$.

Različite metode obučavanja parametara koriste različite informacije, pa se neke fokusiraju samo na najverovatnije tagove, dok druge koriste samo marginalne verovatnoće. U svakom slučaju, važno je da algoritmi za rešavanje ovih problema budu što efikasniji. To je moguće kod linearnih i nekih specijalnih oblika generalnih uslovnih slučajnih polja, dok se u najopštijem slučaju koriste apsoksimacije. Sledeća dva poglavlja bave se ovim algoritmima, i nekim od metoda obučavanja parametrara, kako za linearna, tako i za generalna uslovna slučajna polja .

2.7.2 Viterbi i unapred-unazad algoritmi za linearna uslovna slučajna polja

Viterbi i unapred-unazad su efikasni algoritmi dinamičkog programiranja koji se koriste kod skrivenih Markovljevih modela. Njihove varijante za uslovna slučajna polja mogu se izvesti postupkom sličnim onom kojim smo sa jednog modela prešli na drugi u poglavlju 2.5. Algoritam unapred-unazad je zapravo identičan, jedino se faktori definušu drugačije. Prisetimo se da model uslovnih slučajnih polja možemo da predstavimo kao proizvod faktora

(63) (64). Za ovako definisane faktore, rekurentne jednačine u algoritmima unapred, unazad i Viterbi mogu da se koriste neizmenjene za model linearnih uslovnih slučajnih polja. Jedina razlika je što vrednosi α, β i δ više nemaju značenje koje su ranije imali, kako se sada više ne radi o skrivenim stanjima i generisanju sekvenci, ali ih i dalje definišemo jednakostima (39), (43) i (47). Takođe, rezultat algoritama unapred i unazad nije više verovatnoća sekvence $p(\boldsymbol{x})$ već faktor normalizacije $Z(\boldsymbol{x})$, koji nam je zapravo i potreban u procesu obučavanja parametara. Imajući ovo na umu, marginalne verovatnoće, takođe potrebne za obučavanje parametara, se dobijaju kao:

$$p(y_{t-1}, y_t|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \alpha_{t-1} \Psi_t(x_t, y_t, y_{t-1}) \beta_t(y_t) \tag{74}$$

$$p(y_t|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \alpha_t \beta_t(y_t) \tag{75}$$

Više o ovim algoritmima može se naći u [14].

2.7.3 Zaključivanje kod generalnih uslovnih slučajnih polja

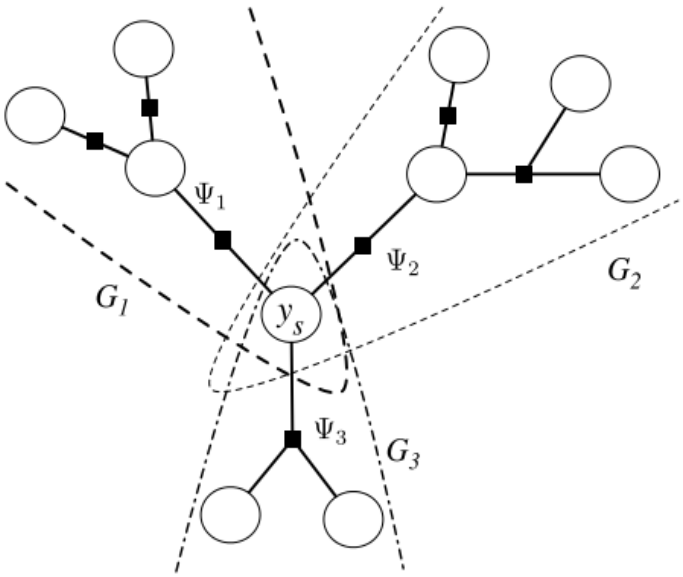
Algoritmi za rad sa generalnim slučajnim poljima, i grafičkim modelima uopšte, su dosta složeniji usled veće slobode definisanja karakterističnih funkcija. Egzaktni algoritmi zaključivanja su u najgorem slučaju eksponencijalne složenosti, iako se u praksi ona često ne dostiže. Najpoznatiji ovakav algoritam, takozvani algoritam "spajanja stabla" (engl. *junction tree algorithm*) transformiše faktor graf u stablo, nakon čega se željene marginalne verovatnoće mogu egzaktno izračunati algoritmima specifičnim za stabla [4]. Više o ovakvim metodama može se naći u [15]. Zbog velike složenosti egzaktnih algoritama, razvijen je veliki broj aproksimacija. Najpoznatije dve klase ovakvih algoritama su Monte Karlo i varijacioni algoritmi. Ovi algoritmi nisu karakteristični samo za uslovna slučajna polja, već i za bilo koji model čija se raspodela verovatnoća faktoriše po nekom faktor grafu, bilo da se radi o $p(y)$ ili $p(x|y)$ [4]. Radi lakše notacije, u nastavku će se zanemariti zavisnost od promenljivih x, i posmatrati samo $p(y)$, za koju smatramo da se faktoriše po faktor grafu $G=(V, F)$ kao

$$p(y) = \frac{1}{Z} \prod_{a \in F} \Psi_a(y_a) \quad . \tag{76}$$

U ovom poglavlju predstavljen je varijacioni algoritam zaključivanja koji pretpostavlja da je faktor graf G stablo - širenje verovanja (engl *belief propagation*). U pitanju je generalizacija

metoda korišćenih za linearna uslovna slučajna polja. Više o drugim metodama može se naći u [4]. Ukoliko dati faktor graf nije stablo, ova metoda se i dalje može koristiti kao aproksimacija, ali nije garantovano da će konvergirati.

Neka je $G=(V, F)$ stablo, a Y_s promenljiva čiju marginalnu verovatnoću želimo da izračunamo. Ideja algoritma je da svaki od faktora susednih Y_s utiče multiplikativno na verovatnoću njoj dodeljene vrednosti, slanjem poruke koja se za svaki faktor može izračunati nezavisno od ostalih, i koja pokazuje koliko je vrednost dodeljena Y_s kompatibilna sa nekim delom grafa. Proizvod svih poruka koje pristignu u Y_s je proporcionalan njenoj marginalnoj verovatnoći. Neka je je $N(s)$ skup svih faktora susednih sa Y_s i neka faktor $a \in N(s)$. Označimo sa $G_a = (V_a , F_a)$ podgraf G koji se nalazi "uzvodno" od Ψ_a . To znači da sve promenljive koje Ψ_a razdvaja od Y_s pripadaju V_a , a svi takvi faktori pripadaju F_a. Ova situacija je ilustrovana slikom 8.



Slika 8, širenje verovanja, preuzeta iz [4]

Skupovi $V_a \setminus Y_s$ su međusobno disjunktni, što je slučaj i sa skupovima Ψ_a. Zbog ovoga je važno da je G stablo, i verovatnoća se može faktorisati kao

$$p(y_s) \propto \sum_{\mathbf{y}/y_s} \prod_{a \in F} \Psi_a(\mathbf{y}_a) = \sum_{\mathbf{y}/y_s} \prod_{a \in N(S)} \prod_{\Psi_b \in F_a} \Psi_b(\mathbf{y}_b) = \prod_{a \in N(S)} \sum_{\mathbf{y}\, v_a/y_s} \prod_{\Psi_b \in F_a} \Psi_b(\mathbf{y}_b) \quad . \tag{77}$$

Ovim jednakostima dobili smo organizaciju faktora koja odgovara prethodno uočenoj podeli na podgrafove. Označimo deo desne jednakosti specefičan za svaki faktor a, odnosno podgraf G_a kao

$$m_{as}(y_s) = \sum_{\mathbf{y} \, v_a / y_s} \prod_{\Psi_b \in F_a} \Psi_b(\mathbf{y_b}) \quad . \tag{78}$$

$m_{as}(y_s)$ je zapavo poruka koja stiže u čvor Y_s iz faktora F_a, i koja govori u uticaju podgrafa G_a na traženu marginalnu verovatnoću. Slično, definišimo poruku koja promenljiva šalje faktoru

$$m_{sa}(y_s) = \sum_{\mathbf{y} \in V_s} \prod_{\Psi_b \in F_a} \Psi_b(\mathbf{y_b}) \quad . \tag{79}$$

Iz jednakosti (77) i (78) sledi da je

$$p(y_s) \propto \prod_{a \in N(S)} m_{as}(y_s) \quad , \tag{80}$$

$$p(\mathbf{y_a}) \propto \Psi_a(\mathbf{y_a}) \prod_{a \in N(a)} m_{sa}(\mathbf{y_a}) \quad . \tag{81}$$

Izrazi (80) i (81) se, radi efikasnoti izračunavanja, mogu rekurzivno izraziti i kao

$$m_{as}(y_s) = \sum_{\mathbf{y_a} / y_s} \Psi_a(\mathbf{y_a}) \prod_{t \in N(a) - \{s\}} m_{ta}(y_t) \quad , \tag{82}$$

$$m_{sa}(y_s) = \prod_{b \in N(s) - \{a\}} m_{bs}(y_s) \quad . \tag{83}$$

Dakle, poruka koju čvor s šalje faktoru a je zapravo proizvod poruka pristiglih u s od svih drugih susednih faktora b. Svaka od tih poruka nosi informaciju o tome koliko je y_s dobro uklopljena u odnosu na fakor b. U poruci koja se šalje faktoru a, ne posmatra se poruka primljena od istog tog faktora, jer bi se u suprotnom ona prenaglašavala. Slično, poruka koju faktor a šalje promenljivoj s je mišljenje tog faktora o vrednosti y_s, pomnoženo sa porukama pristiglim u a, i sumirano po svim dodelama koje promenljivoj Y_s daju vrednost y_s. U prvom koraku, sve poruke se inicijalizuju na 1. U svakom sledećem, šalju se od korena, kako bi se prvo primile poruke od kojih zavise druge, sve do nekog kriterijuma konvergencije. Može se izračunati i marginalna verovatnoća skupa $p(y_a)$, kao

$$p(\mathbf{y_a}) = k \, \Psi_a(\mathbf{y_a}) \prod_{s \in N(s)} m_{sa}(y_s) \quad , \tag{84}$$

pri čemu je k faktor normalizacije, koji služi da suma svih disjunktnih verovatnoća bude 1.

Verovatnoća cele sekvence $\boldsymbol{y}$ može se izraziti kao

$$p(\boldsymbol{y})=\prod_{s\in V} p(y_s)\prod_{a}\frac{p(\boldsymbol{y}_a)}{\prod_{t\in a} p(y_t)} \tag{85}$$

pri čemu se dobija da je

$$Z=\frac{1}{p(\boldsymbol{y})}\prod_{a\in F}\Psi_a(\boldsymbol{y}_a) \quad . \tag{86}$$

Algoritam širenje verovanja primenjen na linearna uslovna slučajna polja je zapravo algoritam unapred-unazad. Više o metodama zaključivanja za generalna uslovna slučajna polja može se naći u [4].

2.8 Obučavanje parametara

2.8.1 Slučaj linearnih uslovnih slučajnih polja

Kao što je u poglavlju 2 napomenuto, procena parametara za uslovna slučajna polja koristi princip maksimalne verodostojnosti. Neka je dat skup za obučavanje $\{\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)}\}_{i=1,\ldots,N}$, pri čemu ćemo zbog lakše sintakse, bez umanjenja opštosti, pretpostaviti da su sve sekvence dužine T. Prikažimo logaritamsku uslovnu verodostojnost (20) jezikom uslovnih slučajnih polja (61) [4]

$$l(w)=\sum_{i=1}^{N}\log p(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)};w)=\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K} w_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, \boldsymbol{x}_t^{(i)})-\sum_{i=1}^{N}\log Z(\boldsymbol{x}^{(i)}) \quad . \tag{87}$$

Kako bismo izbegli problem overfitinga, malo ćemo modifikovati prethodni izraz

$$l(w)=\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K} w_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, \boldsymbol{x}_t^{(i)})-\sum_{i=1}^{N}\log Z(\boldsymbol{x}^{(i)})-\sum_{k=1}^{K}\frac{w_k^2}{2\sigma^2} \quad . \tag{88}$$

Izraz koji je dodat je zapravo euklidska norma vekrota *w*, kombinovana sa faktorom regularizacije $1/2\sigma^2$. σ je slobodan parametar koji treba proceniti, pre svega na osnovu veličine skupa za obučavanje. U ovom slučaju se radi o L2 regulaizaciji. Ukoliko koristimo 1-normu vektora *w*, radi se o L1 regularizaciji:

$$l(w)=\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K} w_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, \boldsymbol{x}_t^{(i)})-\sum_{i=1}^{N}\log Z(\boldsymbol{x}^{(i)})-\alpha\sum_{k=1}^{K}|w_k| \tag{89}$$

gde je, analogno, potreno proceniti parametar α. Kako je

$$\frac{\partial l}{\partial w_k}[\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K} w_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, \boldsymbol{x}_t^{(i)})]=\sum_{i=1}^{N}\sum_{t=1}^{T} f_k(y_t^{(i)}, y_{t-1}^{(i)}, \boldsymbol{x}_t^{(i)}) \tag{90}$$

i

$$\begin{aligned}\frac{\partial l}{\partial w_k}[\sum_{i=1}^{N}\log Z(\boldsymbol{x}^{(i)})] &= \sum_{i=1}^{N}\frac{1}{Z(\boldsymbol{x}^{(i)})}\sum_{y,y'}[\exp\sum_{t=1}^{T}\sum_{j=1}^{K} w_j f_j(y,y',\boldsymbol{x}^{(i)})]\sum_{t=1}^{T} f_k(y,y',\boldsymbol{x}_t^{(i)}) \\ &= \sum_{i=1}^{N}\sum_{y,y'}[\sum_{t=1}^{T} f_k(y,y',\boldsymbol{x}_t^{(i)})]\frac{\exp\sum_{t=1}^{T}\sum_{j=1}^{K} w_j f_j(y,y',\boldsymbol{x}_t^{(i)})}{\sum_{y'',y'''}\exp\sum_{t=1}^{T}\sum_{l=1}^{K} w_l f_l(y'',y''',\boldsymbol{x}_t^{(i)})} \\ &= \sum_{i=1}^{N}\sum_{y,y'}\sum_{t=1}^{T} f_k(y,y',\boldsymbol{x}_t^{(i)})\,p(y,y'|\boldsymbol{x}_t^{(i)})\end{aligned}, \tag{91}$$

to je parcijalni izvod *l(w)* po nekom parametru w_k je

$$\frac{\partial l}{\partial w_k}=\sum_{i=1}^{N}\sum_{t=1}^{T} f_k(y_t^{(i)}, y_{t-1}^{(i)}, \boldsymbol{x}_t^{(i)})-\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{y,y'} f_k(y,y',\boldsymbol{x}_t^{(i)})\,p(y,y'|\boldsymbol{x}_t^{(i)})-\frac{w_k}{\sigma^2}\quad . \tag{92}$$

Intuitivno, izraz (92) govori da je najbolji izbor parametara onaj koji izjednačuje očekivanje sume vrednosti karakterisitčnih funkcija na osnovu datog skupa za obučavanje, i na osnovu tagova predviđenih modelom. Pri računanju ovih vrednosti koriste se algoritmi iz prethodnog poglavlja koji određuju marginalne raspodele i faktor normalizacije $Z(\boldsymbol{x}^{(i)})$. U svakom računanju verodostojnosti, marginalne verovatnoće se računaju za svaku sekvencu $\boldsymbol{x}^{(i)}$, pa je efikasnost algoritama zaključivanja od velikog uticaja na složenost izračunavanja.

Funkcija *l(w)* se u praksi optimizuje numeričkim metodama. Povoljna okolonost je da je to zapravo konkavna funkcija, što znači da je svaki lokalni ekstremum ujedino i globalni. Štaviše, ukoliko je definisana izrazom (88), odnosno ako je korišćena L-2 regularizacija, funkcija je strogo konkavna, te ima samo jedan ekstremum. Neke od numeričkih metoda pogodnih za njenu optimizaciju su Njutnova metoda, varijanta kvazi-Njutnove metode BFGS [14], ili metoda najbržeg spusta [6]. Složenost ovakve optimizacije je $O(TM^2NG)$, gde je $O(TM^2)$ složenost algoritma unapred-unazad koji se koriste pri svakoj iteraciji, N veličina skupa za obučavanje, a G broj iteracija konretne numeričke metode.

2.8.2 Metoda stohastičkog gradijentnog spusta

Metoda opisana u prethodnom poglavlju posmatra ceo skup za obučavanje pre nego što modifikuje parametre u svakoj iteraciji. Alternativa ovakvom pristupu je da nakon svake osmotrene sekvence iz skupa za obučavanje promenimo parametre na osnovu te sekvence, što je jako korisno ako je skup za obučavanje velik, što radi metoda stohastičkog gradijentnog spusta. Posmatrajmo samo jednu sekvencu *(x, y)* iz skupa za obučavanje, i, radi preglednosti, uvedimo oznake karakterističnih funkcija višeg nivoa (65). Tada, analogno računanju parcijalnog izvoda po paramteru w_k (92) za ceo skup za obučavanje, ovog puta radi jednostavnosti bez faktora regularizacije, dobija se izraz

$$\begin{aligned}\frac{\partial l}{\partial w_k}\log p(y|x) &= \sum_{t=1}^{T} f_k(y_t, y_{t-1}, x) - \sum_{y'}\sum_{t=1}^{T} f_k(y'_t, y'_{t-1}, x)\, p(y'|x) \\ &= F_k(x,y) - \sum_{y'} F_k(x,y')\, p(y'|x) \\ &= F_k(x,y) - E_{y'\sim p(y'|x)}[F_k(x,y')]\end{aligned} \quad . \tag{93}$$

I dalje treba imati na umu da ovaj izraz zapravo znači da kada sumiramo jednakosti (93) za ceo skup za obučavanje, želimo da vrednosti karakterističnih funkcija na skupu za obučavanje bude jednak vrednostima koje je model proizveo.

Metoda stohastičkog gradijentnog spusta radi tako što iz skupa za obučavanje izdvoji jedan par sekvecni *(x, y)* na slučajan način. Zatim, za svaku karakterističnu fukciju višeg nivoa (ili analogno nižeg nivoa) F_k, izračuna očekivanje vrednosti ove funkcije na osnovu modela, i modifikuje odgovarajući parametar w_k prema formuli [8]

$$w_k = w_k + \alpha\left(F_k(x,y) - E_{y'\sim p(y'|x)}[F_k(x,y')]\right) \tag{94}$$

Računanje traženog očekivanja radi se numerički [14], ali je vremenski dosta zahtevno. Metoda *Collins perceptron* je modifikacija gradijentnog spusta, koja prevazilazi ovaj problem tako što očekivanje uopšte i ne računa. Štaviše, ova metoda jedino koristi rezultat Viterbi algoritma u procesu obučavanja parametara, tako što parametre modifikuje prema formuli

$$w_k = w_k + \alpha\left(F_k(x,y) - F_k(x,y')\right) \tag{95}$$

gde je y' tag koji je Viterbi algoritam dao kao najverovatniji, a y tag iz skupa za obučavanje. Ako je y različito od y', model se informiše o tome smanjujući vrednost parametra w_k, koji je doveo do nepravilne predikcije. Ako je predikcija tačna, nema potrebe za promenom

parametara.

2.8.3 Slučaj generalnih uslovnih slučajnih polja

Logaritamska uslovna verodostojnost generalnih slučajnih polja može se izraziti kao [4]

$$\begin{aligned} l(w) &= \log p(\boldsymbol{y}|\boldsymbol{x}) = \log\left[\frac{1}{Z}(\boldsymbol{x}) \cdot \prod_{C_p \in C} \prod_{\Psi_c \in C_p} \exp\left[\sum_{k=1}^{K} w_{ak} f_{ak}(x_a, y_a)\right]\right] \\ &= \prod_{C_p \in C} \prod_{\Psi_c \in C_p} \prod_{k=1}^{K(p)} w_{kp} f_{kp}(\boldsymbol{x}_c, \boldsymbol{y}_c) - \log Z(\boldsymbol{x}) \end{aligned} \tag{96}$$

gde nismo sumirali logaritam verovatnoće po svim sekvencama, kao u sličaju linearnih uslovnih slučajnih polja, već se posmatra samo jedna takva sekvenca. Parcijalni izvod po parametru w_{pk} je

$$\frac{\partial l}{\partial w_{pk}} = \sum_{\Psi_c \in C_p} f_{pk}(\boldsymbol{x}_c, \boldsymbol{y}_c) - \sum_{\Psi_c \in C_p} \sum_{\boldsymbol{y}'_c} f_{pk}(\boldsymbol{x}_c, \boldsymbol{y}_c)\, p(\boldsymbol{y}'_c|\boldsymbol{x}) \tag{97}$$

I u slučaju generalnih uslovnih slučajnih polja, funckija logaritamske verodostojnosti je konkavna, pa se i njene ekstremne vrednosti traže numeričkim metodama, kao što su metoda konjugovanih pravaca ili L-BFGS. Radi bolje efikasnoti, razvijen je veliki broj aproksimativnih metoda [16].

3. Rastavljanje reči na kraju reda

Problem rastavljanja reči na kraju reda javlja se pri svakoj pripremi teksta bilo za štampanje, bilo za prikaz na nekom medijumu, i potreban je program koji ga rešava automatski. Jako često se dešava da napisana reč ne može cela da stane u red sa prethodnim rečima, pa ju je poželjno podeliti na dva dela, oh kojih će jedan ostati u tom redu, a drugi se preneti u novi. Na kraju prvog reda stavlja se simbol '-'. Pravila kojima se ovakava podela vrši definisana su pravopisom, ali se zapravo u većini slučajeva mogu koristiti i pravila za rastavljanje reči na slogove, koje se mogu naći u gramatikama. Ovaj problem sam po sebi nije specifičan ni za jedan konkretan prirodni jezik, ali mu se metode kojima se rešava uglavnom prilagođavaju.

3.1 Istorijat

U srpskom jeziku su rastavljanje reči na kraju reda i podela reči na slogove, odnosno silabifikacija, usko povezani problemi. Pojednostavljeno, moglo bi se reći da se na kraju reda reč deli na granici dva sloga. Slučaj sa nekim jezicima, poput engleskog, nije toliko jednostavan, jer bi ovako podeljena reč mogla da prouzrokuje potpuno drugačiji izgovor od željenog, pošto ponekad kasniji slogovi utiču na izgovor prethodnih. Primer ovoga se može videti u izgovorima reči "ne-gation", "ne-onatology", ili "ne-utrality".

Prvi programi za automatsku silabifikaciju engleskog jezika nastali su ranih 60-ih godina, i korišćeni su u novinskim agencijama. Oni su problemu pristupali na osnovu baze unapred pravilno podeljenih reči. Te reči bi znali da podele bez greške, ali nijednu više. Reči koje se nisu nalazile u bazi su se heuristički delile nakon trećeg, petog ili sedmog slova [22]. Problem sa ovakvim pristupom je što je neophodno obezbediti što veću bazu reči - što nosi veliku vremensku i memorijsku složenost u toku izvršavanja, i što za reči van baze softver ne nudi ispravan odgovor. To takođe znači da bi baza trebalo redovno da se dopunjava novim rečima, rečima stranog porekla, itd. Nakon engleskog, sledeći jezik za koji je razvijen program za rastavljanje reči na slogove je finski, 1964. godine [17]. Prvi algoritam za silabifikaciju patentiran je 1964. godine [18]. Ovaj algoritam takođe koristi određenu bazu, posmatra dužinu reči, i prepoznaje određene šablone, na osnovu kojih statistički određuje podelu na slogove. Softver koji je danas najviše u upotrebi, korišćen i u sistemu TEX i njegovim derivatima, nastao je 1983. godine [19].

Različite metode mašinskog učenja su primenjene na problem silabifikacije. Njihova preciznost se može izmeriti na više načina. Kao što ćemo videti u poglavlju 4, predikcija tagova se u slučaju silabifikacije radi na nivou slova, onosno, svakom slovu se dodeljuje tag.

Zbog toga se preciznost modela često meri određenom metrikom u odnosu na svaki tag pojedinačno. Sa druge strane, kako je reč sekvenca slova kojoj se dodeljuje sekvenca tagova, ima smisla posmatrati i preciznost modela u odnosu na broj korektno tagovanih reči. U prvom slučaju kažemo da se preciznost predikcije meri na nivou slova, dok se u drugom slučaju radi o preciznosti na nivou reči. 1995. godine isprobane su tri metode [20], koje su bile precizne za, u proseku, 97% na nivou slova, na osnovu skupa za obučavanje od 100000 reči. Upotreba SVM metode [21] dala je dobre rezultate - 95.65% preciznost na novou reči, kao i upotreba uslovnih slučajnih polja [22], gde je preciznost iznosila 96.33% na nivou reči, odnosno 99.16% na nivou slova, na skupu za obučavanje veličine oko 66000 reči. Uslovna slučajna polja, kao i metoda SVM korišćena je i za problem u rumunskom jeziku [23]. Za srpski jezik postoji program RAS [24] koji služi za rastavljanje reči na kraju reda. Razvijena je metoda automatskog rastavljanja reči srpskohrvatskog jezika na slogove za dva skupa pravila - pravila u pravopisu A. Belića iz 1934. i pravopisu M. Stefanovića, iz 1964. godine, pri čemu se "postupak za određivanje granice sloga zasniva, u opštem slučaju, na određivanju požaja susednih centralnih fonema u reči i ispitivanju konsonanata koji se među njima, eventualno, nalaze"[25]. Program uzima u obzir ova dva skupa pravila i dat skup prefiksa. Razvijena je i metoda za rastavljanje reči na kraju reda u srpskohrvatskom jeziku, koja osim pravila i skupa prefiksa koristi i rečnik izuzetaka [26].

3.2 Pravila za rastavljanje reči na kraju reda u srpskom jeziku

Pravopis srpskog jezika [27] propisuje pravila za ispravnu podelu reči na kraju reda:

a. Ne prenosi se jedno slovo (makar to bio i vokal koji čini slog), jer se ono može ispisati umesto crtice. Nije, znači, dobro: misa-o, ispisa-o, ču-o i sl.
b. Rastavljeni delovi reči (smešteni u dva reda) moraju imati najmanje po jedan slog, što istovremeno znači i da se ne rastavljaju jednosložne reči bez obzira na broj slova: plast, sklad, vlast...
c. Ako je na mestu preloma postojala (polusloženička) crtica, onda se jedna crtica piše na kraju a druga na početku narednoga reda: auto- -salon, korak- -dva, pet- -šest, i sl.
d. U latinici se ne razdvajaju dvojna slova (lj, nj, dž) u značenju jednog glasa: vo-lja, pa-žnja, ho-dža. Ovog se valja držati i prilikom pisanja stranih imena.
e. Kad se nađe suglasnik između dva vokala, on pripada delu reči koji se prenosi u sledeći red: va-ljati, de-vojka, o-vako...
f. Ako se na kraju reda deli suglasnička grupa, kraj prvog dela treba da odgovara mogućem ('prirodnom') kraju a početak prenesenog dela mogućem početku reči, odnosno - sastavi suglasničkih grupa i kraja i početka rastavljenih delova moraju biti u skladu sa uobičajenim rasporedom glasova u srpskom jeziku. Neki primeri ovog praivla dati su u tabeli 1.

g. Kad prelom složenica na kraju reda padne oko spoja njihovih prepoznatljivih i smisaono prozirnih sastavnica, treba ih deliti granicom sastavnica: iz-baciti, iz-graditi, od-voditi i sl. Ako je na spoju sastavnica došlo do uprošćavanja suglasničkih grupa ispadanjem, uprošćeni dvojni suglasnik pripada drugoj sastavnici: be-stidan. Ako je umetnuto slovo a, ono pripada drugoj sastavnici: raza-gnati. Međutim, ako je granica delova složenice zamagljena i neuočljiva, podelu treba vršiti prema opštim pravilima: o-teti, ra-ljutiti, u-zeti i sl.

ispravno	neispravno	neispravno
brat-stvo	brats-tvo	bra-tstvo
brat-ski	brats-ki	bra-tski
grad-ski	grads-ki	gra-dski
umrt-vljen	umrtvlj-en	umr-tvljen
tram-vaj	tamv-aj	tra-mvaj
mač-ji	ma-čji	
maj-ka	ma-jka	
dovolj-no	dovo-ljno	
par-tija	pa-rtija	

Tabela 1, primeri pravila f

Takođe, data je sledeća napomena: "*Mada za pažljivu podelu reči na kraju reda ima poneki specifičan detalj (tradicionalno nasleđen), neće se pogrešiti ako se primene pravila podele reči na slogove, kakva se mogu naći u gramatikama srpskog jezikika.*

Pored ovih pravila, u starijim pravopisima [28] postoje i druge preporuke, poput pravila da se ne ostavlja jedan vokal u prvom redu. Ipak, i pored svih ovih pravila, ostavljena je određenja količina slobode pri podeli.

Neka od datih pravila, kao što je pravilo *a*, je lako implementirati. Međutim, pojedina pravila, pogotovo *d, f* i *g*, nije tako lako izraziti. Pre svega, računar ne može da zna da li je "nj" jedno ili dva slova za reči koje nisu date u bazi. Da bi mogao da prepozna "prirodni" početak reči potrebno je da posmatra date prefikse reči, što nije efikasno. Na kraju, kako ne bi prekršio pravilo *g*, morao bi da zna semantiku date reči. Ovo je zato što pravopis preferira semantičku podelu reči na slogove. Sa druge strane, gramatika srpskog jezika [29] daje pravila za podelu reči na slogove koja je fonetskog karaktera. Ona daje "opis sloga kao glasovne jedinice u jeziku", posmatrajući slova po njihovoj prirodi. Za veliki broj reči, ova dva pristupa daju isti rezultat, ali postoje i izuzeci, prikazani u tabeli 2.

fonetska podela	ra-zlju-ti-ti	o-du-ze-ti	i-ste-ra-ti
semantička podela	raz-lju-ti-ti	od-u-ze-ti	is-te-ra-ti

Tabela 2, primeri fonetske i semantičke podele reči

Radi poređenja prirode fonetske i semantičke podele reči na slogove, navedimo neka od pravila koja nalaže gramatika:

> *Uz opšte pravilo da se granica sloga nalazi iza samoglasnika, a ispred suglasnika, normativna gramatika propisuje i sledeća pravila.*
>
> *1) Kada se u unutrašnjosti reči nađe više suglasnika, odnosno sonanata, od kojih je na prvom mestu neki strujni ili sliveni suglasnik, granica sloga će biti ispred te grupe suglasnika: la-sta, vo-ćka, ma-čka...*
> *2) Ispred suglasničke grupe biće granica sloga i ako se u grupi suglasnika u unutrašnjosti reči na dugom mestu nalazi neki od sonanata v, j, r, l ili lj, a ispred njega bilo koji drugi suglasnik osim sonanata: to-pla, do-bra, sve-tlost...*
> *3) Ako grupu u unutrašnjosti reči čine dva sonanta, granica sloga dolazi između njih, pa jedan sonant pripada prethodnom sloga, a drugi sledećem: tram-vaj, mar-ljiv, in-va-lid...*
>
> ...

Jasno je da ova dva skupa pravila drugačije pristupaju problemu, iako u većini slučajeva, dolaze do istog zaključa. Ipak, gramatička pravila su dosta stroža, i ne ostavljaju prostor za dvoumljenje.

4. Primena linearnih slučajnih polja na rastavljanje reči na kraju reda u srpskom jeziku

Sledeći korak je predstavljanje problema rastavljanja reči na kraju reda jezikom uslovnih slučajnih polja. Konkretnije, za ovaj problem ćemo razviti model linearnih uslovnih slučajnih polja, koji za proizvoljnu reč **x** treba da predvidi pozicije slova iza kojih se reč sme rastaviti.

4.1 Kodiranje promenljivih

Pre svega, potrebno je predstaviti promenljive u modelu. Neka je sekvenca **x** reč, za koju želimo da odredimo tag **y**. Karakteristike **x** koje će nam biti od interesa su njegovi karakteri. Ako je dužina ove reči T, definišimo **y** kao binarnu sekvencu iste dužine, tako da je i-ti karakter taga 1 ako se reč **x** može rastaviti nakon i-tog karaktera, a 0 u suprotnom. Nekoliko ovakvih sekvenci prikazano je u tabeli 3.

x	**y**
misao	01000
oduzeti	0110100
isterati	01010100

Tabela 3, primeri sekvenci x i y

Kako linearna uslovna slučajna polju mogu da posmatraju proizvoljne karakterisike pri proceni svakog taga (61), umesto pojedinačnih karaktera možemo da posmatramo i proizvoljne podniske reči, ili čak celu reč. Ako se prisetimo algoritama za zaključivanje iz poglavlja 2.7.2, primetićemo da je mala kardinalnost skupa iz kojeg tagovi uzimaju vrednosti, što je svakako slučaj sa binarnim skupom, jako povoljna osobina modela.

4.2 Karakteristične funkcije

Kao karakteristike date sekvence **x**, radi predikcije taga $\mathbf{y}_t$, posmatraćemo podniske **x** dužina 2, 3 i 4 karaktera, koje sadrže $\mathbf{x}_t$. Na primer, ako je data sekvenca "enciklopedija", i ukoliko želimo da procenimo odgovarajući tag za slovo "o", koje se nalazi na poziciji t=7, vektor karakteristika koji nam je od interesa je sačinjen od sledećih niski $\overline{\mathbf{x}}$ = {"lo", "op",

"klo", "lop", "ope", "iklo", "klop", "lope", "oped"}. Pri svakoj predikciji se posmatraju i prethodni tagovi. Ilustrujmo detaljno na ovom primeru kako bi neke karakteristične funkcije izgledale, i koje je njihovo značenje.

Funkcija f_k je indikatorska funkcija, koja je definisana nekim šablonom, i za konkretne vrednosti svojih argumenata predstavlja indikator događaja da je dati šablon prepoznat. Na primer, dafinišimo jednu takvu funkciju kao

$$f_1(x, y_t, y_{t-1}, t) = I(x_{t-1}x_t x_{t+1} = \text{lop}, y_{t-1} = 0, y_t = 1) \tag{98}$$

gde smo argumentima funkcije dodali i poziciju t, radi preglednosti zapisa (inače se podrazumeva da je pozicija poznata). Uz nju se definišu i propratne funkcije za različite vrednosti tagova

$$\begin{aligned} f_2(x, y_t, y_{t-1}, t) &= I(x_{t-1}x_t x_{t+1} = \text{lop}, y_{t-1} = 0, y_t = 0) \\ f_3(x, y_t, y_{t-1}, t) &= I(x_{t-1}x_t x_{t+1} = \text{lop}, y_{t-1} = 1, y_t = 0) \\ f_4(x, y_t, y_{t-1}, t) &= I(x_{t-1}x_t x_{t+1} = \text{lop}, y_{t-1} = 1, y_t = 1) \end{aligned} \tag{99}$$

Ovakvih funkcija može da ima jako mnogo, i one imaju višestruku ulogu u modelu. Pre svega, potrebni su nam odgovarajući parametri, koji se računaju na osnovu skupa za obučavanje algortimima predstavljenim u poglavlju 2.8.1.

Neka je u skupu za obučavanje data sekvenca "enciklopedija" i njoj odgovarajuća sekvenca tagova "0101001010100". Posmatrajmo karaktere na poziciji t=7, i izračunajmo vrednosti prethodno definisanih funkcija:

$$\begin{aligned} f_1(\text{enciklopedija}, 1, 0, 7) &= 1 \\ f_2(\text{enciklopedija}, 1, 0, 7) &= 0 \\ f_3(\text{enciklopedija}, 1, 0, 7) &= 0 \\ f_4(\text{enciklopedija}, 1, 0, 7) &= 0 \end{aligned} \tag{100}$$

Takođe, vrednosti svih ovih funkcija za bilo koju drugu poziciju t, i odgovarajuće tagove y_t i y_{t-1} je 0:

$$f_k(\textit{enciklopedija}, y_t, y_{t-1}, t) = 0, \quad k \in \{1,2,3,4\}, t \neq 7 \tag{101}$$

Posmatrajući samo ovu reč, zaključujemo da f_1 najbolje opisuje skup za obučavanje, i treba joj dodeliti najveći parametar. Međutim, konkretnim parametrima nije od značaja na kojoj poziciji u sekvencama je karakteristična funkcija dala potvrdan odgovor, jer se u izrazu (61) vrši sumiranje po svim pozicijama t. U tom smislu je preglednije posmatrati karakteristične funkcije višeg nivoa (65), koje govore koliko puta je odgovarajuća karakteristična funkcija

prepoznala svoj šablon za date sekvence **x** i **y**. Tako je

$$\begin{aligned} F_1(enciklopedija, 0101001010100) &= 1 \\ F_k(enciklopedija, 0101001010100) &= 0, \quad k \in \{2,3,4\} \end{aligned} \tag{102}$$

Pretpostavimo sada da je u skupu za obučavanje data reč "klopka" i njen tag "000100", i želimo da izračunamo vrednosti funckija za poziciju t=3:

$$\begin{aligned} f_1(klopka, 0, 0, 3) &= 0 \\ f_2(klopka, 0, 0, 3) &= 1 \\ f_3(klopka, 0, 0, 3) &= 0 \\ f_4(klopka, 0, 0, 3) &= 0 \end{aligned} \tag{103}$$

odnosno

$$\begin{aligned} F_2(klopka, 000100) &= 1 \\ F_k(klopka, 000100) &= 0, \quad k \in \{1,3,4\} \end{aligned} \tag{104}$$

Sada kada imamo malo bolji uvid u skup za obučavanje, vidimo da funkcijama f_1 i f_2 treba dodeliti nešto veće parametre od funkcija f_3 i f_4. Ako bismo naišli i na reč "lopta" i njegov tag "00100", ponovo bismo prednost dali funkciji f_2, i tako dalje. Svrha obučavanja parametara je dodeljivanje težina definisanim karakterističnim funkcijama, na osnovu toga koliko dobro one oslikavaju skup za obučavanje.

Odavde je jasno da je ukupan broj mogućih karakterističnih funkcija jako velik. Kako srpska azbuka ima 30 slova, broj ovakvih funkcija definisanih za podniske reči dužine 3 bi bio $4*30^3=108000$. Za dužinu podniske 5 ovaj broj bi iznosio 97200000. Međutim, veliki broj kombinacija slova, kao što je "ppp", se gotovo nikada ne javlja u prirodnom jeziku. Zbog toga je poželjno konstruisati karakteristične funkcije prema onim kombinacijama koje se javljaju, pre svega, u skupu za obučavanje. U ovom radu su, kao što je napomento, posmatrane sve podniske dužina 2, 3 i 4 koje su se javile u skupu za obučavanje, i prema njima konstruisane karakteristične funckije na prethodno opisan način.

U fazi testiranja, potrebno je proceniti tag sekvence koja nije do tada viđena. To se, kako je napomenuto u poglavlju 2.7.2. radi specifičnim algoritmom dinamičkog programiranja, koji ponovo na sličan način računa vrednosti definisanih funkcija, i odabira tag koji ove funkcije na određeni način odabiraju kao podesan.

Na isti način konstruisani su i drugi skupovi karakterističnih funkcija, odnosno drugi modeli uslovnih slučajnih polja, koji posmatraju podniske dužine 2, 3, 4 ili 5, kao i neke njihove

kombinacije, kako bi se došlo do najboljeg mogućeg izbora. Ispostavljeno je da je najbolji skup funkcija upravo onaj koji posmatra podniske dužina 2, 3 i 4.

4.3 Skup za obučavanje i skup za testiranje

Da bi konstruisani model mogao da proceni parametre, potrebno je snabdeti ga skupom za obučavanje, odnosno skupom reči i njihovim ispravnim tagovima. Korišćena su dva ovakva skupa. Prvi skup čini nešto više od 4000 različitih ručno podeljenih reči. Drugi skup ima oko 31000 različitih reči podeljenih programom RAS [24]. U oba slučaja, 80% reči pripada skupu za obučavanje, a ostatak skupu za testiranje. Ova podela je bitna kako bi se procenila preciznost predikcije na rečima koje model nije video u procesu obučavanja, jer upravo ona oslikava preciznost u budućoj upotrebi.

U svakom od ova dva slučaja, preciznost se meri na osnovu date podele. To znači da je, za drugi skup sekvenci, verovatnoća da preciznost razvijenog modela nadmaši RAS-ovu preciznost jako mala, jer on zapravo uči da oponaša RAS. Čak i da ova prednost postoji, ona ne bi bila vidljiva bez dodatne, objektivne, ručno generisane podele skupova za obučavanje i testiranje.

Pravila korišćena za "ispravnu" podelu reči u ova dva skupa se donekle razlikuju. Tako bi, na primer, u prvom skupu mogla da se nađe podela "u-či-ni-ti", dok bi u drugom ona glasila "uči-ni-ti", ili kombinacija reči "ma-čka" i "mač-ka". Zanimljivo je posmatrati kako se model prilagođava ovakvim razlikama. Sa druge strane, iz istog razloga se model treniran na jednom skupu ne može testirati u odnosu na drugi. Međutim, u tome upravo leži velika prednost metoda mašinskog učenja nad metodama definisanim pravilima - opremite model podacima, i on će se snaći. Primetimo da nigde pri konstruisanju modela nismo posmatrali ikakve specifičnosti srpskog jezika ili pravila koja nalažu pravopis i gramatika. Te stvari su bitne pri kreiranju skupova za obučavanje i testiranje, ali ako su oni poznati, model ne mora ništa više da zna. To između ostalog znači da se isti model može primeniti na rešavanje problema silabifikacije u engleskom, nemačkom ili italijanskom jeziku.

4.4 CRF++

Postoji više nekomercijalnih softverskih alata koji implementiraju uslovna slučajna polja. U ovom radu korišćen je program CRF++, pisan u C++ jeziku, i dostupan na adresi [30]. Još neki od ovih programa su CRFsuite, CRFSGD, SRFSharp... Svi oni funkcionišu tako što primaju unapred generisane datoteke, na osnovu kojih formiraju karakteristične funkcije i vrše predikciju. Za upotrebu CRF++ potrebno je formirati tri datoteke: skup za obučavanje,

skup za testiranje, i datoteku šablona, koje treba da zadovolje određeni format. Pravila za njihovo generisanje mogu se naći na adresi programa.

Skup za obučavanje i skup za testiranje zapravo treba da budu u istom formatu, i nazivaju se datoteke podataka. Svaki red datoteke je jedan token, sačinjen od vektora karakteristika jedne reči (odnosno elementa čiji se tag određuje), iza kojih se nalazi njegov ispravan tag. Ove karakteristike mogu biti sama ta reč, neka njegova osobina, prefiks, i sl, i razdvojeni su blanko karakterima. Svaka sekvenca predstavljena je nizom tokena, dok su dve sekvence razdvojene praznim redom. Primer ovakve datoteke, preuzet sa stranice programa [30] date je u tabeli 4.

```
He          PRP     B-NP
reckons     VBZ     B-VP
the         DT      B-NP
current     JJ      I-NP
account     NN      I-NP
deficit     NN      I-NP
will        MD      B-VP
narrow      VB      I-VP
to          TO      B-PP
only        RB      B-NP
#           #       I-NP
1.8         CD      I-NP
billion     CD      I-NP
in          IN      B-PP
September   NNP     B-NP
0           0       O

He          PRP     B-NP
reckons     VBZ     B-VP
```

Tabela 4, primer datoteke podataka

U ovom primeru radi se o predikciji imenovanih entiteta. Data je jedna sekvenca reči "He reckons the current account deficit will narrow to only # 1.8 billion in September.", i druga koja počinje sa "He reckons". Ove reči smeštene su u prvu kolonu, sa razmakom između dve rečenice. U drugoj koloni nalaze se vrste tih reči, koje su unapred poznate. Same reči i njihove vrste su karakteristike. U trećoj koloni navedeni su korektni tagovi.

Datoteka šablona ima nešto kompleksniji format. Svaki red ove datoteke predstavlja jedan šablon na osnovu koga će se konstruisati karakeristične funkcije i može se predstaviti regularnim izrazom [UB]\d+:\%x\[-?\d+,\d+\](/\%x\[-?\d+,\d+\])* (zanemareni su blanko karakteri), na primer "U01:%x[0,0]". Prvi deo šablona je njegov identifikator. Makro %x[*kolona*,*red*] se koristi kako bi se iz skupa za obučavanje izvukli potrebni podaci i na osnovu njih konstruisala karakteristična funkcija. Broj *kolona* specizira relativnu poziciju u skupu za obučavanje u odnosu na red tokena koji se trenutno čita, dok je *red* apsolutna pozicija kolone. Program radi tako što prolazi kroz skup za obučavanje, sekvencu po sekvencu. U svakom trenutku on posmatra jedan red ove datoteke, koji ćemo nazivati trenutni token. Za svaki od trenutnih tokena prolazi se kroz celu datoteku šablona koji se proširuju konkretnim vrednostima. Datoteka šablona za malopređašnji problem data je u tabeli 5.

```
# Unigram
U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]
U05:%x[-1,0]/%x[0,0]
U06:%x[0,0]/%x[1,0]

U10:%x[-2,1]
U11:%x[-1,1]
U12:%x[0,1]
U13:%x[1,1]
U14:%x[2,1]
U15:%x[-2,1]/%x[-1,1]
U16:%x[-1,1]/%x[0,1]
U17:%x[0,1]/%x[1,1]
U18:%x[1,1]/%x[2,1]

U20:%x[-2,1]/%x[-1,1]/%x[0,1]
U21:%x[-1,1]/%x[0,1]/%x[1,1]
U22:%x[0,1]/%x[1,1]/%x[2,1]

# Bigram
B
```

Tabela 5, primer datoteke šablona

Pretpostavimo da se trenutno u skupu za obučavanje posmatra token "`the DT B-NP`".
Tada prvi šablon označava događaj da je reč koja se nalazi dva reda iznad trenutnog tokena

baš "He". Za svaki tag y' pronađen u skupu za obučavanje formira se po jedna karakteristična funkcija oblika "reč pre prethodne je "He" i trenutni tag je y' ". Neka je broj različitih tagova L. Drugi šablon, za isti posmatrani trenutni token, formira L funkcija oblika "prethodna reč je "reckons" i trenutni tag je y' ". Šabloni kojima je vrednost *kolona* 1, umesto reči posmatraju vrste reči, ali sve ostalo je isto, sve dok je šablon imenovan prefiksom 'U'. Ovo su unigram šabloni, koji posmatraju proizvoljne karakteristike tokena i trenutni tag. Ukoliko želimo da posmatramo i prethodni tag, što uglavnom jeste slučaj, potrebni su nam bigram šabloni, koji počinju slovom 'B'. Ako slovo B stoji samo, formiraju se dodatni bigrami od svih unigram šablona. Ako je N broj broj jedinstvenih niski kreiranih nekim unigram šablonom, broj ukupnih funkcija koje on definiše je L*N, dok je u slučaju bigram šablona taj broj L*L*N.

Zamislimo da su definisani šabloni U31:%x[2,1] i U32:%x[3,1]. Ako je trenutni token koji se posmatra `"the DT B-NP"`, ova ova šablona definišu isto pravilo za vrstu reči, NN. Identifikacija šablona se zato vrši i numerički, kako bi se ove dve karakteristike NN razlikovale. Sami brojevi, osim toga, ne igraju nikakvu ulogu - dok god se međusobno razlikuju.

Kada je forimaran skup za obučavanje *train_file* i datoteka šablona *template_file*, obučavanje parametara i kreiranje modela se poziva iz komandne linije pozivom *crf_learn* programa

% crf_learn template_file train_file model_file

gde je *model_file* novogenerisani model, koji se može evaluirati na skupovima za testiranje *test_files* sledećom komandom

% crf_test -m model_file test_files ...

Program *crf_test* će na standardni izlaz ispisati rezultat predikcije, koji se lako može preusmeriti u novu datoteku. Rezultat je u sličnom formatu kao datoteke podataka, pri čemu je kreirana još jedna kolona - predviđeni tag. Format rezultirajuće datoteke dat je u tabeli 6.

Preciznost predikcije se lako može odrediti željenom metrikom na osnovu vrednosti u pretposlednjoj i poslednjoj koloni, odnosno pravog i predviđenog taga.

Programi *crf_learn* i *crf_test* mogu imati i dodatne argumente, o kojima se više može naći na adresi [30].

```
Rockwell        NNP     B     B
International   NNP     I     I
Corp.           NNP     I     I
's              POS     B     B
Tulsa           NNP     I     I
unit            NN      I     I
```

Tabela 6, primer rezultata predikcije

4.5 Implementacija

Na osnovu oba skupa reči formirane su datoteke podataka, odnosno skup za obučavanje i skup za testiranje. Skup za testiranje u prvom slučaju iznosi 17% ukupnih reči, dok u drugom iznosi 20%. Kao što je opisano u poglavlju 4.1, reči se zapravo posmatraju kao sekvence slova, a mogući tagovi su 0 i 1. Kako kao karakteristike reči posmatramo slova, svaki token u datoteci podataka je zapravo par (slovo, tag), dok su reči razdvojene praznim redom. Format datoteka podataka je dat u tabeli 7

```
i       0
z       0
a       1
b       0
r       0
a       1
t       0
l       0
```

Tabela 7, primer datoteke podataka

Korišćeni su bigram šabloni koji posmatraju podniske reči dužina 2, 3 i 4. Datoteka šablona je data u tabeli 8.

```
B01:%x[-1,0]/%x[0,0]
B02:%x[0,0]/%x[0,1]
B10:%x[-2,0]/%x[-1,0]/%x[0,0]
B11:%x[-1,0]/%x[0,0]/%x[1,0]
B12:%x[0,0]/%x[1,0]/%x[2,0]
B20:%x[-3,0]/%x[-2,0]/%x[-1,0]/%x[0,0]
B21:%x[-2,0]/%x[-1,0]/%x[0,0]/%x[1,0]
B12:%x[-1,0]/%x[0,0]/%x[1,0]/%x[2,0]
```

Tabela 8, korišćeni šabloni

Pretpostavimo da posmatramo sekvencu slova datu tabelom 7, i da je slovo koje trenutno čitamo 'b'. Šablon B11 tada definiše događaje "Prethodno slovo je 'a', trenutno slovo je 'b', sledeće slovo je 'r' i trenutni tag je '0'" i "Prethodno slovo je 'a', trenutno slovo je 'b', sledeće slovo je 'r' i trenutni tag je '1'".

Formirane su i dodatne datoteke šablona, kako bi se uporedila preciznost različitih modela. Ovi šabloni su posmatrali podniske dužina 2, 3, 4 ili 5, i neke kombinacije ovih podniski. Ispostavilo se da je najprecizniji model onaj čiji su šabloni definisani upravo tabelom 8.

4.6 Rezultati

Vrednost svakog pojedinačnog taga može biti 0 ili 1, i on može biti predviđen ispravno ili ne. Korišćene su tri metrike koje ocenjuju tačnost modela u odnosu na pojedinačno slovo, i jedna u odnosu na cele reči.

Promenljive koje se koriste za ove tri metrike su TP, TN, FP i FN. TP je broj tačno predviđenih jedinica, TN je broj tačno predviđenih nula, dok su FP i FN brojevi netačno predviđenih jedinica, odnosno nula. Na osnovu njih definišu se sledeće vrednosti

$$\begin{aligned} accuracy &= \frac{TP+TN}{TP+TN+FP+FN} \\ precision &= \frac{TP}{TP+FP} \\ recall &= \frac{TP}{TP+FN} \end{aligned} \tag{105}$$

Svaka od ovih metrika na određeni način izražava uspešnost modela. Kako je pogrešno predviđena jedinica veći problem u rastavljanju reči na kraju reda od pogrešno predviđene nule, korišćene su metrike *precision* i *recall* koje se fokusiraju upravo na taj slučaj, kao i generalna metrika *accuracy*. Preciznost predikcije u odnosu na reč je jednostavno broj usprešno predviđenih sekvenci tagova za celu reč u odnosu na ukupan broj reči. Ovaj broj je obično manji od preciznosti u odnosu na slovo, jer je dovoljno da je samo jedno slovo pogrešno tagovano da bi cela reč bila pogrešna. U tabelama 9 i 10 predstavljenosti su ove vrednosti za oba skupa za testiranje, kao i vremena izvršavanja učenja parametara.

U prvoj koloni tabela su date dužine podniski reči koje su posmatrane kao karakteristike. Model je očigledno lošije radio za prvi skup za testiranje, zbog male veličine odgovarajućeg skupa za obučavanje. Najveća dostignuta tačnost po reči je 92.839%, dok je preciznost po slovima u najboljem slučaju 98.851%. Kod drugog skupa za testiranje ovi brojevi su 97.53%

i 99.57%, što je svakako bolji rezultat.

dužina	accuracy (%)	precision (%)	recall (%)	po reči (%)	vreme (s)
2	98.795	97.642	98.538	92.839	0.7
3	98.556	97.792	97.652	90.864	0.71
4	97.912	97.222	96.145	86.79	0.89
5	96.72	96.293	93.221	80.124	1.24
2, 3, 4	98.851	98.484	97.873	92.839	1.4

Tabela 9, preciznost na skupu za testiranje koji sadrži 20% od 4000 reči

dužina	accuracy (%)	precision (%)	recall (%)	po reči (%)	vreme (s)
2	98.896	97.399	98.767	93.049	6.98
3	99.393	98.817	99.059	96.738	4.43
4	99.23	98.466	98.842	95.583	6.13
5	98.588	97.585	97.456	92.749	8.66
2, 3, 4	99.57	99.117	99.38	97.53	9.59

Tabela 10, preciznost na skupu za testiranje koji sadrži 20% od 31000 reči

Preciznost modela se određuje na osnovu skupa za testiranje. Međutim, korisno je testirati model i u odnosu na skup za obučavanje, kako bi se videlo koliko su parametri dobro prilagođeni. U tabeli 11 prikazani su rezultati testiranja modela koji posmatraju podniske dužine 2, 3 i 4 za oba skupa za obučavanje.

skup	accuracy (%)	precision (%)	recall (%)	po reči (%)
1	99.841	99.808	99.656	99.135
2	99.987	99.977	99.975	99.929

Tabela 11, preciznost na skupovima za obučavanje

U ovom slučaju preciznost je skoro jednaka, što ukazuje na problem overfittinga za prvi skup. Međutim, ovde se taj problem može rešiti jedino dodavanjem novih reči.

Vreme izvršavanja obučavanja parametara se u svakom od modela meri u sekundama, što je jako efikasno. Ovo je uglavnom posledica malog broja mogućih tagova. Model koji posmatra samo podniske dužine 3 se pokazao kao najprecizniji za drugi skup reči, pored modela koji posmatra podniske dužine 2, 3 i 4, ali i nešto brži. Za prvi skup reči precizan je model koji posmatra podniske dužine 2. U tom smislu, u zavisnosti od konkretnog problema, može se razmišljati o kompromisima između brzine i preciznosti. Međutim, brzina koja je važna pri upotrebi programa je zapravo brzina testiranja, koje je u svakom od slučajeva bilo trenutačno.

5. Zaključak

U ovom radu predstavljena su uslovna slučajna polja, diskriminativni probabilistički model. Prikazan je odnos sa drugim diskriminatvinim i generativnim modelima, i na osnovu njih su razvijeni algoritmi za rad sa sekvencama. Algoritmi za obučavanje parametara nasleđeni su od modela logističke regresije, dok su algoritmi zaključivanja skrivenih Markovljevih polja prilagođeni za linearna uslovna slučajna polja.

Opisan je problem rastavljanja reči na kraju reda u sprkom jeziku, neki od načina dosadašnjeg rešavanja, i navedena su pravila koja on treba da zadovoljava.

Konstruisan je model linearnih uslovnih slučajnih polja za podelu reči na kraju reda u srpskom jeziku. Skup od 31000 reči dobijenih softverom RAS podeljen je na skup za obučavanje i skup za testiranje u odnosu 80:20. Korišćen je softver CRF++. Na ovom skupu za testiranje model je dostigao preciznost od 97.53% na novou reči, odnosno 99.57% na nivou slova. Zadatak koji bi trebalo rešiti u budućnosti je manuelno generisanje skupa za obučavanje i testiranje kako bi razvijen model mogao da se uporedi sa postojećim metodama za podelu reči na kraju na reda, kao što je pomenuti softver RAS.

6. Reference

[1] J. Lafferty, A. McCallum, F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data" *International Conference on Machine Learning (ICML)*, 2001.

[2] P. Le-Hong, P. Xuan-Hieu i T. The-Trung, "On the effect of the label bias problem in part-of-speech tagging", *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, 2013.

[3] J. Lasserre, C. M. Bishop, "Generative or Discriminative? Getting the Best of Both Worlds", *Bayesian statistics, Vol. 8*, 2007.

[4] C. Sutton, A. McCallum, "An Introduction to Conditional Random Fields", *Foundations and Trends in Machine Learning Vol. 4, No. 4,* 2011.

[5] F. R. Kschischang, B. J. Frey i H. Loeliger, "Factor Graphs and the Sum-Product Algorithm", *IEEE transactions on information theory, Vol. 47, No. 2*, 2001

[6] D. Radunović, "Numeričke metode", *Građevinska knjiga, Beograd*, 1991.

[7] D. Jurafsky i J. H. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics", 2nd edition, *Prentice-Hall*, 2009.

[8] C. Elkan, "Log-linear models and conditional random fields", I*n UCSD CSE250B Lecture Note*, 2013.

[9] A. L. Berger, S. A. Pietra, V. J. Pietra, "A maximum entropy approach to natural language processing", C*omputational Linguistics,* 1996

[10] R. Durbin, S. Eddy, A. Krogh, i G. Mitchison, "Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids", *Cambridge University Press*, 1998.

[11] G. L. Kouemou, "istory and Theoretical Basics of Hidden Markov Models, Hidden Markov Models, Theory and Applications, Dr. Przemyslaw Dymarski (Ed.)", *InTech,* 2011.

[12] A. McCallum, D. Freitag, F. Preira, "Maximum Entropy Markov Models for Information Extraction and Segmentation", *ICML '00 Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

[13] Y. Liu, E. Shriberg, A. Stolcke i M. Harper, "Comparing HMM, Maximum entropy, and Conditional Random Fileds for Disfluency Detection", *In Proceeedings of the European Conference on Speech Communication and Technology,* 2005.

[14] R. Gupta, "Conditional Random Fields", *Unpublished report, IIT Bombay*, 2006.

[15] D. Koller i N. Friedman, "Probabilistic Graphical Models: Principles and Techniques" , *MIT Press*, 2009.

[16] C. A. Sutton, "Efficinet training methods for conditional random fileds", *Ph.D. thesis, University of Massachusetts,* 2008.

[17] T. Jarvi, "Computerized Typesetting and Other New Applications in a Publishing House", *IFIP Advances in Information and Communication Technology*, 2009.

[18] F. J. Damerau. 1964. "Automatic Hyphenation Scheme", *U.S. patent 3537076*, 1964
[19] F. M. Liang, "Word Hy-phen-a-tion by Com-put-er", *Ph.D. thesis, Stanford University*, 1983.
[20] A.Bosch , T.Weijters , J. Herik i W. Daelemans, "The Profit of Learning Exceptions", *In Proceedings of the 5th Belgian-Dutch Conference on Machine Learning,* 1995.
[21] S. Bartlett , G. Kondrak i C. Cherry, "Automatic Syllabification with Structured SVMs for Letter-To-Phoneme Conversion", *Proceedings of ACL-08: HLT* , 2008.
[22] N. Trogkanis i C. Elkan, " Conditional random fields for word hyphenation", ACL 2010 *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden, pages 366–374. The Association for Computer Linguistics*, 2010.
[23] L. P. Dinu, V. Niculae i O. Sulea, "Romanian Syllabication using Machine Learning", *Lecture Notes in Computer Science Volume 8082*, 2013.
[24] Stranica programa: http://www.rasprog.com
[25] D. Vitas, "Podela na slogove srpskohvatskih reči", *Informatica 3*, 1981
[26] C. Krstev, "Rastavljanje reči srpskohrvatskog jezika na kraju retka", Z*bornik radova sa III naučnog skupa "Računarska obrada jezičkih podataka",* 1985.
[27] M. Pešikan, "Pravopis srpskog jezika", *Matica srpska*, 2008.
[28] M. Pešikan, "Pravopis srpskog jezika", *Matica srpska*, 1993.
[29] Ž. S. Stanojčić, "Gramatika srpskog književnog jezika", *Kreativni centar*, 2010.
[30] T. Kudo, "CRF++: Yet Another CRF Toolkit", web stranica programa: http://crfpp.googlecode.com/svn/trunk/doc/index.html