P(izraz | gramatika)

Urh Primožič
Mentor: prof. dr. Ljupčo Todorovski
Somentor: asist. dr. Matej Petković

28. februar 2022

V jezikoslovju pojem gramatika označuje sistem jezikovnih sredstev, ki definira slovnična pravila jezika. Z uporabo teh pravil povezujemo znake in besede v smiselna besedila. Poljubnemu jeziku pripada neskončna množica smiselnih povedi.

V matematiki besedilo predstavimo z nizom znakov iz abecede, slovnična pravila pa nadomestimo s strukturami, ki tvorijo množico besedil. Primer take strukture je verjetnostna gramatika, ki poleg pravil skladnje definira verjetnosti za posamezne besede in besedila.

Znake iz abecede lahko nadomestimo z matematičnimi simboli in raziskujemo gramatike, ki podajajo pravila za ustvarjanje matematičnih izrazov. Osrednja tema diplomskega dela je algoritem, ki pri podani verjetnostni gramatiki izračuna verjetnost za poljuben matematičen izraz.

1 Kontekstno neodvisne gramatike

V diplomskem delu so vse gramatike kontekstno neodvisne. V splošnem poleg takih poznamo širši razred kontekstno odvisnih gramatik. V spodnjih definicijah za poljubno množico V z V^* označimo množico vseh končnih zaporedij iz V. Elementom $(a_1, \ldots, a_n) \in V^*$ rečemo besede in pišemo $a_1 \cdots a_n \in V^*$.

Definicija 1. *Naj bosta N in T poljubni disjunktni, neprazni, končni množici simbolov in naj bo v množici N odlikovan začetni simbol $S \in N$. Naj bo $R \subseteq N \times (N \cup T)^*$ celovita relacija.*

To definira kontekstno neodvisno gramatiko $G = (N, T, S, R)$. Množici $V = N \cup T$ rečemo abeceda, simbolom iz T pa končni simboli. Urejene pare iz R imenujemo prepisovalna pravila. Par $(A, \alpha) \in R$ pišemo kot $A \to \alpha \in R$ in rečemo, da se A prepiše v α. Rečemo tudi, da se zaporedje simbolov $\alpha_1 A \alpha_2 \in V^*$ prepiše v $\alpha_1 \alpha \alpha_2$ in pišemo $\alpha_1 A \alpha_2 \to \alpha_1 \alpha \alpha_1$.

Definicija 2. *Gramatika $G = (N, T, S, R)$ tvori besedo $w \in T^*$, če obstaja končno zaporedje $(a_i)_{i=1}^n$ elementov množice V^*, da je $a_1 = S$, $a_n \in T^*$ in se a_i prepiše v a_{i+1} za vsak $i \in \{1, \ldots, n-1\}$.*

Množico vseh besed, tvorjenih z gramatiko G označimo z $L(G)$ in ji rečemo jezik gramatike G.

Definicija 3. *Vsaki besedi $w \in L(G)$ pripada izpeljevalno drevo, ki poda postopek tvorbe besede. Izpeljevalno drevo je orientirano drevo z vozlišči iz množice V, za katero velja*

- *koren drevesa je znak S.*
- *otroci nekončnega simbola $A \in N$ so točno elementi neke besede $\alpha \in V^*$, v katero se prepiše A. V drevesu so urejeni od leve proti desni v enakem vrstnem redu, kot nastopajo v α,*
- *listi drevesa so natanko končni simboli,*

Vsakemu notranjemu vozlišču drevesa s simbolom $A \in N$ ustreza eno prepisovalno pravilo $A \to \alpha = X_1 X_2 \dots X_n$. V tem primeru ima vozlišče n naslednikov, ki ustrezajo simbolom $X_i, i = 1 \dots n$.

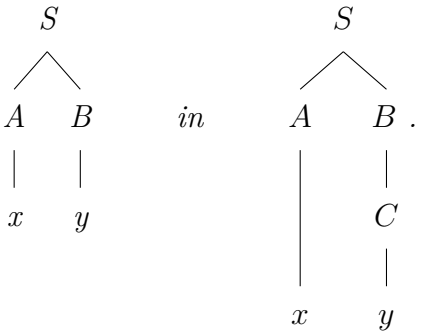
Vsako drevo poda postopek za tvorjenje besede $w = a_1 \cdots a_n \in L(G)$, kjer so $a_1, \dots, a_n$ listi drevesa, spisani od leve proti desni. Velja pa tudi obratno: za vsako besedo $w \in L(G)$ obstaja drevo, ki ustreza kombinaciji prepisovalnih pravil, ki prepišejo začetni simbol S v besedo w.

Množico vseh dreves gramatike G označimo z $\Psi(G)$, z $B(\tau)$ pa besedo, ki jo izpelje drevo τ.

Zgled 1. *Naj bo $G = (\{S, A, B, C\}, \{x, y\}, S, R)$ gramatika s prepisovalnimi pravili*

$$r_1 = S \to AB$$
$$r_2 = A \to x$$
$$r_3 = B \to y$$
$$r_4 = B \to C$$
$$r_5 = C \to y$$

Gramatika tvori le besedo xy. Edini drevesi v gramatiki sta



Če je jasno, katera prepisovalna pravila so uporabljena, drevesa pišemo linearno. Prvo drevo lahko zapišemo kot

$$S \xrightarrow{r_1} AB \xrightarrow{r_2} xB \xrightarrow{r_3} xy.$$

Če hočemo poleg besed v jeziku gramatike definirati še verjetnostno porazdelitev nad njimi, uporabljamo verjetnostne kontekstno neodvisne gramatike [2]. Slednje prepisovalnim pravilom priredijo verjetnosti, na osnovi katerih lahko računamo verjetnosti izpeljevalnih dreves in nato še verjetnosti besed.

Definicija 4. *Naj bo $G = (V, T, S, R)$ kontekstno neodvisna gramatika in $P\colon R \to [0, 1]$ preslikava, da za vse nekončne simbole $A \in N$ in vsa prepisovalna pravila $A \to w \in R$ velja*

$$\sum_{(A \to w) \in R} P(A \to w) = 1$$

Definicija 5. *Verjetnost izpeljevalnega drevesa $\tau \in \Psi(G)$ definiramo kot produkt verjetnosti vseh pravil, ki se pojavijo v τ, torej*

$$P(\tau) = \prod_{r \in R} P(r)^{f_G(r)}$$

kjer je $f_G(r)$ število pojavitev pravila r v drevesu τ.

Definicija 6. *Naj bo $w \in L(G)$. Definiramo verjetnost za besedo $w \in L(G)$ kot*

$$P(w) = \sum_{B(\tau) = w} P(\tau)$$

2 Tvorjenje enačb

V množici končnih simbolov so lahko spremenljivke $x_1, \ldots, x_n$, binarni operatorji ($+$, $-$, $*$, $/$ ...), oklepaji, vejice in simbole za funkcije (sin, cos, exp ...). Potem lahko opazujemo družino gramatik, ki tvori smiselne matematične izraze.

Zgled 2. *Opazujmo spodnjo gramatiko s končnimi simboli $+, -, y$ in x.*

$$\begin{aligned} S &\to S + V \mid S - V \mid V \\ V &\to x \mid y \mid xy \end{aligned}$$

Vse besede iz jezika te gramatike predstavljajo izraze oblike $ax + bxy + cy$ za neka cela števila a, b, c.

Verjetnostne gramatike, ki tvorijo enačbe, so koristne pri simbolni regresiji, tj. algoritmičnem iskanju enačb, ki se prilegajo podatkom . Algoritem iz [1] s pomočjo verjetnostne gramatike tvori kandidatke za enačbe, med katerimi izbere tisto, pri kateri je odstopanje vrednosti enačbe od meritev najmanjše. Pri tem je ločevanje enačb, ki se razlikujejo le v konstantah, nesmiselno, saj lahko konstante izberemo tako, da se določena oblika enačbe podatkom čim bolj prilega. Algoritmi zato uporabljajo gramatike, ki tvorijo enačbe z nedoločenimi konstantami.

Zgled 3. *Naj bo G gramatika s končnimi simboli c, x in $+$ in praviloma $S \to S + cx \mid cx$. Vse besede so oblike $cx + \cdots + cx$. Algoritem za simbolno regresijo vzame besedo $cx + cx$, ki predstavlja enačbo $Ax + Bx$, kjer sta A in B prosti konstanti. Vrednosti A in B algoritem izbere tako, da se enačba $Ax + Bx$ najbolj prilega podatkom glede na mero za napako. Naj bosta ti vrednosti A_0 in B_0. Ker velja $Ax + Bx = (A + B)x$, je $A_0 + B_0$ tista vrednost spremenljivke, pri kateri se enačba, ki jo predstavlja beseda cx, najbolj prilagaja podatkom. Opazimo, da so z vidika izbiranja konstant vse besede ekvivalente, saj se enačbe, ki jih dobimo iz njih, da zapisati kot Ax. Z vidika simbolne regresije vse besede te gramatike predstavljajo enak matematični izraz.*

Algoritem za svoje delovanje potrebuje verjetnost za posamezen izraz, ki jo trenutno oceni. Cilj diplomskega dela je razvoj točnega algoritma za izračun verjetnosti za dan izraz. Za to je nujna točna matematična definicija izraza, ki zaobjame lastnosti, izpostavljene v zgledu.

Definicija 7. *Naj bo G gramatika z množico končnih simbolov*

$$T = \{c, x_1, \ldots, x_n, (,)\} \cup O \cup F$$

kjer je O množica izbranih matematičnih operatorjev in ločil ter F množica simbolov za funkcije. Naj G tvori izključno besede, ki imajo nedvoumen matematični pomen, ko spremenljivkam $x_1, \ldots, x_n$ in konstantam c priredimo neke vrednosti.

Naj bo $w \in L(G)$ beseda, ki jo tvori taka gramatika. Pišemo jo kot $w = w_1 c w_2 c \cdots c w_{n+1}$, kjer velja, da c ne nastopa v členih w_i. Definiramo preslikavo $\Phi\colon L(G) \to P(\{f\colon U_1 \times \cdots \times U_n \to D \mid U_i \subseteq D\})$

$$\Phi(w) = \{(x_1, \ldots, x_n) \mapsto w_1 c_1 w_2 c_2 \cdots c_n w_{n+1} \mid c_1, \ldots, c_n \in \mathbb{F}\}$$

kjer je domena funkcije $(x_1, \ldots, x_n) \mapsto w_1 c_1 w_2 c_2 \cdots c_n w_{n+1}$ vedno največja možna, da je funkcija še definirana.

V tem delu predpostavimo $D = \mathbb{R}$ in $\mathbb{F} = \mathbb{R}$.

Preslikava Φ besedam priredi vse možne funkcije, ki jih dobimo z različno izbiro vrednosti konstant. Dve besedi sta intuitivno ekvivalentni, če ju Φ preslika v enako množico funkcij.

Definicija 8. *Na $L(G)$ definiram ekvivalenčno relacijo*

$$w \sim v \Leftrightarrow \Phi(w) = \Phi(v)$$

Množica izrazov $I(G) = L(G)/_\sim$ je kvocienta množica relacije $\sim$. Verjetnost izraza $i \in I(G)$ izračunam po formuli

$$P(i) = \sum_{\substack{w \in L(G) \\ \Phi(w) = i}} P(w)$$

3 Rezultati

Podrobneje študiramo cikle gramatike, torej zaporedja prepisovalnih pravil oblike

$$A_1 \to L_1 A_2 D_1 \to \cdots \to L_1 \cdots L_{n-1} A_n D_{n-1} \cdots D_1 \to L_1 \cdots L_n A_1 D_1 \cdots D_1$$

kjer so $A_i \in N$ nekončni simboli, $L_i, D_i \in V^*$ pa zaporedja simbolov. Če so vsi L_i, D_i ničelni (zaporedja 0 elementov), je cikel linearen. Velja izrek

Izrek 1. *Dreves brez ciklov je končno.*

Izkaže se, da iz poljubne gramatike lahko postopoma odstranimo vse linearne cikle in pri tem ne spremenimo verjetnostne porazdelitvne na besedah. Posledično lahko verjetnost katerekoli besede rešimo z rekurzivnim algoritmom.

Verjetnost izraza i lahko računamo z vsoto $\sum_{\Phi(B(\tau))=i} P(\tau)$. Če je dreves, ki izpeljejo izraz i, končno, je algoritem za izračun verjetnosti trivialen. Splošni primer lahko rešimo tako, da vsoto po neskončni množici prevedemo na vsoto po končni množici.

Zgled 4. *Naj bo gramatika G s končnimi simboli $x_1, \ldots, x_n, +, -$ in začetnim simbolom E podana s spodnjimi pravili*

$$\begin{array}{lcl} E & \to & E + cV \;\; [p] \mid c \;\; [1-p] \\ V & \to & x_1 \;\; [q_1] \mid \cdots \mid x_n \;\; [q_n] \end{array}$$

Izrazi, ki jih tvori ta gramatika, so oblike $\{(x_1, \ldots, x_n) \mapsto c_0 + c_1 x_{r_1} + \cdots + c_k x_{r_k}] \mid c_0, \ldots, c_k \in \mathbb{F}\} = \Phi^{-1}([c + cx_{r_1} + \cdots + cx_{r_k}])$. *Velja*

$$\begin{aligned} P([c + cx_{r_1} + \cdots + cx_{r_k}]) & \\ &= \sum_{i=k}^{\infty} \left(\sum_{l_1 + \cdots + l_k = n} \binom{i}{l_1, \ldots, l_k} (1-p) p^i q_{r_1}^{l_1} \cdots q_{r_k}^{l_k} \right) \\ &= \sum_{i=k}^{\infty} \left((1-p) p^i (q_{r_1} + \cdots + q_{r_k})^i \right) \\ &= \frac{1-p}{1 - p(q_{r_1} + \cdots + q_{r_k})} p (q_{r_1} + \cdots + q_{r_k})^{k+1} \end{aligned}$$

Verjetnost za izraz, podan z besedo s k členi cx_i s hitrim potenciranjem lahko izračunamo v $O(k \log k)$.

Splošno gramatiko skušamo razbiti na kombinacijo gramatik, za katere poznamo algoritem.

Literatura

[1] Jure Brence, Ljupčo Todorovski, and Sašo Džeroski. Probabilistic grammars for equation discovery. *Knowledge-Based Systems*, 224:107077, 2021.

[2] Charles S Wetherell. Probabilistic languages: A review and some open questions. *ACM Computing Surveys (CSUR)*, 12(4):361–379, 1980.