Sztuczna Inteligencja

Soma Dutta

Wydział Matematyki i Informatyki, UWM w Olsztynie
soma.dutta@matman.uwm.edu.pl

Wykład - 9: Wnioskowanie w logice I rzędu

Semestr letni 2022

Rachunek zdań: zalety i wady

- Zalety
  - Rachunek zdań jest deklaratywny: elementy syntaktyki odpowiadają faktom.
  - Rachunek zdań dopuszcza częściową/alternatywną/zanegowaną informację (w przeciwieństwie do większości struktur danych i baz danych)
  - Rachunek zdań jest ekstensjonalny: znaczenie $B_{1,1} \wedge P_{1,2}$ wynika ze znaczeń $B_{1,1}$ i $P_{1,2}$.
  - Znaczenie w rachunku zdań jest niezależne od kontekstu (w przeciwieństwie do języka naturalnego)
- Wady
  - Rachunek zdań ma bardzo ograniczoną moc wyrażalności (w przeciwieństwie do języka naturalnego), np. nie da się wyrazić zdania ‘pułapki powodują wiatr na sąsiednich polach’ inaczej niż przez napisanie oddzielnego zdania dla każdego pola

Logika I-go rzędu

- Rachunek zdań zakłada, że świat zawiera tylko fakty, natomiast logika I rzędu (bliższa językowi naturalnemu) zakłada, że świat zawiera:
  - Obiekty: ludzie, domy, liczby, teorie, kolory, baseball, wojny, wieki, ...
  - Relacje: czerwony, okrągły, pierwszy, większy niż, w środku, część, ma kolor, posiada, ...
  - Funkcje: trzeci właściciel, o jeden więcej niż, początek, ...

Syntaktyka (składnia) i semantyka

- Stałe: KingJohn, 0, 1, 2 ...
- Predykaty: Brother, $>$, $=$, ...
- Funkcje: $\sqrt{}$, $+$, ...
- Zmienne: x, y, z, ...
- Spójniki: $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$
- Kwantyfikatory: $\forall$, $\exists$
- Termy:
  - Wszystkie stałe i zmienne są termami.
  - function(term$_1$, term$_2$, ..., term$_n$) jest termem.
  - Na przykład: $3 + 7$, $\sqrt{x}$, ...
- Zdania:
  - Zdania atomowe: predykat(term$_1$, term$_2$, ..., term$_n$). (Na przykład: $\sqrt{x} = 10$. Tutaj warto zauważyć, że zamiast $=(\sqrt{x}, 10)$ jest napisane $\sqrt{x} = 10$.)
  - Zdania złożne są budowane ze zdań atomowych przy pomocy spójników i kwantyfikatorów. (Na przykład: $(\sqrt{x} = 10) \Rightarrow ((x > 3) \wedge \neg(x > 4))$ i $\forall_x(x + (-x) = 0)$)

Semantyka

- Rachunek zdań zakłada, że istnieją fakty, które mają lub nie mają miejsca w świecie (są prawdziwe albo nie w świecie). Każdy fakt może mieć jedną z wartości logicznych: prawda lub fałsz, a każdy model przypisuje wartość prawda lub fałsz do każdego zdania atomowego.
- Logika pierwszego rzędu zakłada więcej; mianowicie, że świat składa się z obiektów o określonych relacjach między nimi, które zachodzą lub nie. Formalne modele są odpowiednio bardziej skomplikowane niż te dla rachunku zdań.
- W logice I rzędu, prawdziwość zdań określa się względem modelu i interpretacji.
  - Model zawiera ≥ 1 obiekt (element dziedziny) i relacje między nimi.
  - Interpretacja specyfikuje przyporządkowania:
    symbole stałych $\rightarrow$ obiekty || predykaty $\rightarrow$ relacje || symbole funkcyjne $\rightarrow$ relacje funkcyjne

- Zdanie atomowe tzn. predicate(term$_1$, ..., term$_n$) jest prawdziwe $\leftrightarrow$ obiekty przyporządkowane do term$_1$, ..., term$_n$ są w relacji przyporządkowanej przez interpretację dla predicate.
- Na przykład, rozważmy zdanie atomowe $\mathbf{x} + \mathbf{3} \preceq \mathbf{11}$ nad dziedziną $\mathbb{N}$. Ta dziedzina ma obiekty ze zbioru $\{0, 1, 2, 3, \ldots\}$, ma relacje fukcyjne $+$, $\times$, i ma relacje $=$, $\leq$.
  - Model: $\mathbb{N} = (\{0, 1, 2, 3, \ldots\}, +, \times, \leq, =)$.
  - Możemy mieć rózne interpretacje. Za pomocą interpretacji $\mathcal{I}$ możemy specyfikować znaczenie różnych symboli w języku.
    - $\mathcal{I}(\mathbf{x}) \in \{0, 1, 2, 3, \ldots\}$,
    - $\mathcal{I}(\mathbf{x} + \mathbf{3}) = \mathcal{I}(\mathbf{x}) + 3$,
    - $\mathcal{I}(\mathbf{x} + \mathbf{3} \preceq \mathbf{11})$ jest prawdziwe jeśli liczba $\mathcal{I}(\mathbf{x} + \mathbf{3})$ jest mniejsza lub równa 11; tzn., jeśli $\mathcal{I}(\mathbf{x})$ jest mniejsza lub równa 8.
- Interpretacje logicznych spójników są jak w przypadku rachunku zdań.

Kwantyfikatory

- Kwantyfikator uniwersalny: $\forall_x S(x)$ gdzie $S(x)$ jest zdaniem zawierające x.
- Na przykład: Każdy w Berkeley jest sprytny: $\forall_x Lives(x, Berkley) \Rightarrow Smart(x)$
- $\forall_x S(x)$ jest prawdziwe w modelu $\mathcal{M}$ w odniesieniu do interpretacji $\mathcal{I}$ wtedy i tylko wtedy gdy interpretacja $S(x)$, czyli $\mathcal{I}(S(x))$ jest prawdziwa dla każdego przyporządkowania x-owi możliwego obiektu w modelu.
- $\forall_x (x^2 \geq 0)$ jest prawdziwe dla wszystkich możliwych wartości, które można przypisać do x przez interpretację $\mathcal{I}$ w dziedzinie liczb rzeczywistych.
- Kwantyfikator egzystencjalny: $\exists_x S(x)$
- Na przykład: Ktoś w Standford jest sprytny: $\exists_x Lives(x, Standford) \wedge Smart(x)$
- $\exists_x S(x)$ jest prawdziwe w modelu $\mathcal{M}$ w odniesieniu do interpretacji $\mathcal{I}$ wtedy i tylko wtedy gdy interpretacja $S(x)$, czyli $\mathcal{I}(S(x))$ jest prawdziwa dla pewnego obiektu przyporządkowanego zmiennej x.

Tłumaczenie zdań języka naturalnego na zdania I rzędu

- Bracia są rodzeństwem: $\forall_{x,y} Brothers(x, y) \Rightarrow Siblings(x, y)$
- Relacja rodzeństwo jest symetryczna:
  $\forall_{x,y} Siblings(x, y) \Leftrightarrow Siblings(y, x)$
- Matka dowolnej osoby jest kobietą i jej rodzicem:
  $\forall_{x,y} Mother(x, y) \Rightarrow (Female(x) \wedge Parents(x, y))$
- Kuzyni są dziećmi rodzeństwa rodziców:
  $\forall_{x,y} Cousin(x, y) \Leftrightarrow$
  $\exists_{z,w} (Parents(z, x) \wedge Parents(w, y) \wedge Siblings(z, w))$

Związki pomiędzy $\forall$ i $\exists$

- $\forall_x Likes(x, IceCreame) \equiv \neg\exists_x \neg Likes(x, IceCream)$
- Ponieważ $\forall$ to tak naprawdę koniunkcja nad przestrzenią obiektów, a $\exists$ to alternatywa, to zachowane są zasady De Morgana.
  - $\forall_x \neg P(x) \equiv \neg \exists_x P(x)$ $\quad (\neg p \wedge \neg q \equiv \neg(p \vee q))$
  - $\neg \forall_x P(x) \equiv \exists_x \neg P(x)$ $\quad (\neg(p \wedge q) \equiv \neg p \vee \neg q))$
  - $\forall_x P(x) \equiv \neg\exists_x \neg P(x)$ $\quad (p \wedge q \equiv \neg(\neg p \vee \neg q))$
  - $\exists_x P(x) \equiv \neg\forall_x \neg P(x)$ $\quad (p \vee q \equiv \neg(\neg p \wedge \neg q))$

Równość

- $\text{term}_1 = \text{term}_2$ jest prawdziwe w danej interpretacji wtedy i tylko wtedy gdy w tej interpretacji termom term_1 i term_2 przyporządkowany jest ten sam obiekt.
- Capital(Poland) = Warsaw
  - term_1 to Capital(Poland) gdzie Capital to symbol funkcyjny i Poland to symbol stałej.
  - term_2 to Warsaw, tzn. term ten jest symbolem stałej.

Świat Wumpusa: baza wiedzy w logice I rzędu

- W przeciwieństwie do rachunku zdań opis bazy wiedzy za pomocą języka logiki pierwszego rzędu jest znacznie bardziej zwięzły.
- Aby opisać bazę wiedzy o świecie Wumpus, w każdym momencie bazującym na odczuciu agenta program musi dodać dodatkową wiedzę do bazy wiedzy.
- Odpowiednie zdanie pierwszego rzędu przechowywane w bazie wiedzy musi wyrażać zarówno percepcję, jak i czas, w którym zdarzenie wystąpiło; w przeciwnym razie agent się zdezorientuje, kiedy co zobaczył.
- Agent w świecie Wumpusa odczuwający zapach i wiatr, ale nie obserwujący błysku w chwili t = 5. Typowym zdaniem percepcyjnym może być $\mathit{Percept}([\mathit{stench}, \mathit{breeze}, \mathit{none}], 5)$, gdzie Percept jest predykatem binarnym zawierającym jako argumenty listę stałych jako pierwszy argument i stałą, konkretną dodatnią liczbę całkowitą, jako drugi argument.

Reguły wnioskowania w logice I rzędu

- Symbole stałych: stench, breeze, glitter, grab, move-forward, move-left, move-right, shoot ... 1, 2, 3, ..., $\text{square}_{1,1}$, ...
- Zmienne: $s, b, g, a, t, sq \ldots$
- Predykaty: $Percept([s, b, g], t)$, $Breeze(t)$, $Stench(t)$, $Glitter(t)$, $Action(t, a)$, $At(sq, t)$, $Pit(sq)$, $Breeze(sq)$ ...
- Biorąc pod uwagę percepcje, można przedstawić takie reguły pierwszego rzędu.
  $\forall_{t,s,g} Percept([s, breeze, g], t) \Rightarrow Breeze(t)$.
  $\forall_{t,b,g} Percept([stench, b, g], t) \Rightarrow Stench(t)$.
  $\forall_{t,s,b} Percept([s, b, glitter], t) \Rightarrow Glitter(t)$.
- $\forall_t Glitter(t) \Rightarrow Action(t, grab)$
- $\forall_{x,y,z,w} Adjacent(square_{x,y}, square_{z,w}) \Leftrightarrow$ $(x = z \wedge (y = w - 1 \vee y = w + 1) \vee (y = w \wedge (x = z - 1 \vee x = z + 1))$
- $\forall_{sq,t} At(sq, t) \wedge Breeze(t) \Rightarrow Breeze(sq)$.
- $\forall_{sq} Breeze(sq) \Leftrightarrow \exists_{sq'} Adjacent(sq, sq') \wedge Pit(sq')$

Wnioskowanie przez podstawienie

- Aby zaprojektować program, możemy podać mu informacje z bazy wiedzy w następujący sposób.
  Tell(KB, Percept([Stench, Breeze, none], 5))
- Aby wywnioskować, jakie działania należy podjąć, system może skierować zapytanie do KB.
  $Ask(KB, \exists_a Action(5, a))$
- Aby zdecydować, KB musi zastąpić zmienną akcji a wszystkimi możliwymi akcjami.
- Możliwe podstawienie:
  $\{a/moveforward, a/moveleft, a/moveright, a/grab, a/shoot \ldots\}$
- Dla danego zdania S i podstawienia σ, $S\sigma$ oznacza wynik zastosowania σ do S.
- Na przykład, jeśli S to $\exists_a Action(5, a)$ i σ oznacza $a/moveforward$, to $S\sigma$ oznacza $Action(5, moveforward)$.
- $Ask(KB, S)$ zwraca pewne/wszystkie σ takie, że $KB \models S\sigma$.

Podstawienia

- Podstawienie: Zbiór postaci $\{x_1/t_1, x_2/t_2, \ldots, x_n/t_n\}$ (1) gdzie $x_1, x_2, \ldots, x_n$ są różnymi zmiennymi, natomiast $t_1, t_2, \ldots, t_n$ są termami.
- Dopuszczamy podstawienie puste, oznaczane ϵ.
- Niech θ będzie podstawieniem postaci (1) i niech α będzie wyrażeniem (tzn. formułą lub termem). Piszemy $\alpha\theta$ na oznaczenie wyrażenia powstałego z α w wyniku jednoczesnego zastąpienia wszystkich wolnych wystąpien zmiennych $x_1, \ldots, x_n$ termami $t_1, \ldots, t_n$.
- Na przykład: Załóżmy, że α jest $Breeze(sq) \Leftrightarrow \exists_{sq'} Adjacent(sq, sq') \wedge Pit(sq')$.
  - sq jest zmienną wolną (free variable), a sq' jest zmienną związaną (bound variable).
  - Jeśli $\theta = \{sq/square_{2,1}\}$, to $\alpha\theta$ ma postać $Breeze(square_{2,1}) \Leftrightarrow \exists_{sq'} Adjacent(square_{2,1}, sq') \wedge Pit(sq')$

Istancjacja uniwersalna

- Każda instancjacja uniwersalnie kwantyfikowanego zdania jest jego logiczną konsekwencją zgodnie z regułą:

$$\frac{\forall_x \alpha(x)}{Subst(\{x/t\}, \alpha(x))}$$

  dla dowolnej zmiennej x i termu ustalonego t.

- Instancjacja uniwersalna może być stosowana wielokrotnie, zeby dodać nowe zdania; po uzyskaniu nowych informacji nowa KB jest logicznie równoważna poprzedniej.
- Na przykład:

$$\frac{\forall_{sq}(Breeze(sq) \Leftrightarrow \exists_{sq'} Adjacent(sq, sq') \wedge Pit(sq'))}{Breeze(square_{2,1}) \Leftrightarrow \exists_{sq'} Adjacent(square_{2,1}, sq') \wedge Pit(sq')}$$

Redukcja do rachunku zdań

Załóżmy, że mamy daną następującą bazę wiedzy KB:

$$\begin{aligned}&\forall x\ King(x) \land Greedy(x) \implies Evil(x)\\&King(John)\\&Greedy(John)\\&Brother(Richard, John)\end{aligned}$$

Instancjując zdania z kwantyfikatorami uniwersalnymi na wszystkie możliwe sposoby otrzymujemy

$$\begin{aligned}&King(John) \land Greedy(John) \implies Evil(John)\\&King(Richard) \land Greedy(Richard) \implies Evil(Richard)\\&King(John)\\&Greedy(John)\\&Brother(Richard, John)\end{aligned}$$

Nowa KB jest sprowadzona do języka zdań: symbole zdaniowe to

$$King(John),\ Greedy(John),\ Evil(John),\ King(Richard),\ \text{itd.}$$

Redukcja do rachunku zdań

Fakt: zdanie ustalone jest logiczną konsekwencją nowej KB

$\iff$

jest logiczną konsekwencją oryginalnej KB

Fakt: każda baza wiedzy bez kwantyfikatorów egzystencjalnych i symboli funkcyjnych może być sprowadzona do języka rachunku zdań, tak aby zachować logiczne konsekwencje

Redukcja do rachunku zdań: efektywność

Redukcja do rachunku zdań generuje wiele nieistotnych zdań.

Np.

$$\forall x\ King(x) \land Greedy(x) \implies Evil(x)$$
$$King(John)$$
$$\forall y\ Greedy(y)$$
$$Brother(Richard, John)$$

Fakt $Evil(John)$ wydaje się oczywisty, ale redukcja produkuje wiele faktów takich jak $Greedy(Richard)$, które są nieistotne.

Dla p k-arnych predykatów i n stałych, będzie $p \cdot n^k$ instancjacji!

Unifikacja

Wniosek $Evil(John)$ z poprzedniego slajdu można otrzymać bardziej bezpośrednio, jeśli znajdziemy podstawienie θ takie, że $King(x)$ oraz $Greedy(x)$ pasują do $King(John)$ oraz $Greedy(y)$.

$\theta = \{x/John, y/John\}$ spełnia te wymagania.

$UNIFY(\alpha, \beta) = \theta$ jeśli $\alpha\theta = \beta\theta$

p	q	θ
$Knows(John, x)$	$Knows(John, Jane)$	$\{x/Jane\}$
$Knows(John, x)$	$Knows(y, OJ)$	$\{x/OJ, y/John\}$
$Knows(John, x)$	$Knows(y, Mother(y))$	$\{y/John, x/Mother(John)\}$
$Knows(John, x)$	$Knows(x, OJ)$	$fail$

Unifikacja

Podstawienie θ jest unifikatorem wyrażeń $E_1, \ldots, E_n$ jeśli $E_1\theta = E_2\theta = \ldots = E_n\theta$.

Podstawienie θ jest najogólniejszym unifikatorem wyrażeń $E_1, \ldots, E_n$ jeśli dla każdego unifikatora tych wyrażeń, γ, istnieje podstawienie λ takie, że $\gamma = \theta\lambda$.

Warto szukać jak najogólniejszych unifikacji.
Należy uważać przy przemianowywaniu zmiennych:
Wszystkie kwantyfikowane zmienne muszą być różne.

Uogólniony Modus Ponens

- Modus ponens w rachunku zdań: $\frac{p, \quad p \Rightarrow q}{q}$.

$$\frac{p_1{}', \; p_2{}', \; \ldots, \; p_n{}', \; (p_1 \land p_2 \land \ldots \land p_n \implies q)}{q\theta} \qquad \text{gdzie } p_i{}'\theta = p_i\theta \text{ dla wszystkich } i$$

$p_1{}'$ jest $King(John)$	p_1 jest $King(x)$
$p_2{}'$ jest $Greedy(y)$	p_2 jest $Greedy(x)$
θ jest $\{x/John, y/John\}$	q jest $Evil(x)$
$q\theta$ jest $Evil(John)$	

Uogólnione Modus Ponens używa baz wiedzy klauzul definiujących (dokładnie jeden literał pozytywny)

Zakłada się, że wszystkie zmienne są kwantyfikowane uniwersalnie

Przykład

Prawo amerykańskie określa, że sprzedaż broni obcemu narodowi przez Amerykanina jest przestępstwem. Państwo Nono, które jest wrogie, ma pewne pociski. Wszystkie pociski zostały mu sprzedane przez pułkownika Westa, który jest Amerykaninem.

Pokazać, że pułkownik West jest przestępcą.

Baza wiedzy: przykład

...sprzedaż broni obcemu narodowi przez Amerykanina jest przestępstwem:

$American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \implies Criminal(x)$

Nono ...ma pewne pociski, tzn. $\exists x\ Owns(Nono, x) \wedge Missile(x)$:

$Owns(Nono, M_1)$ i $Missile(M_1)$

...wszystkie pociski zostały mu sprzedane przez pułkownika Westa:

$\forall x\ Missile(x) \wedge Owns(Nono, x) \implies Sells(West, x, Nono)$

Pociski są bronią:

$Missile(x) \implies Weapon(x)$

Nieprzyjaciel Ameryki uznawany jest za “wrogi”:

$Enemy(x, America) \implies Hostile(x)$

West, który jest Amerykaninem ...

$American(West)$

Państwo Nono, nieprzyjaciel Ameryki ...

$Enemy(Nono, America)$

Dziękuję za uwagę