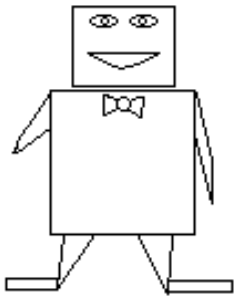
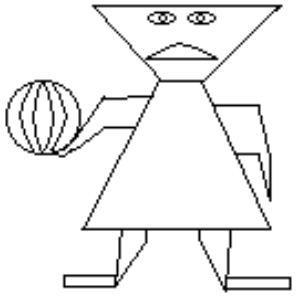
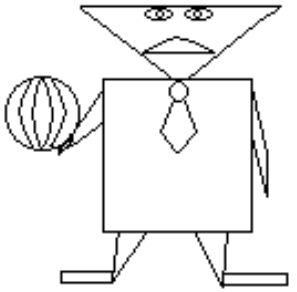
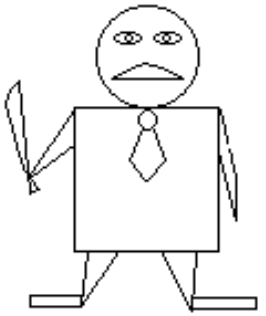
Induktivní logické programování

Příklad 1 „počítačová hra“. Můžeme se naučit roboty rozlišit na základě krátké zkušenosti? *(O. Štěpánková, L. Popelínský)*

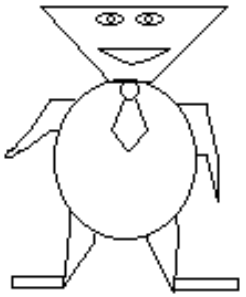
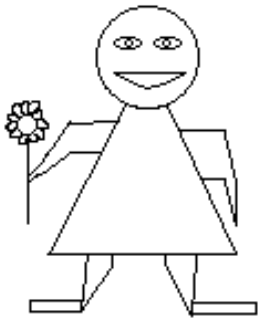



přátelství

nepřátelství



Rozhodovací strom vs. logický prog.

- Rozhodovací strom provádí pouze testy na atributy, např. Hlava=kruh,
  - lze přepsat do pravidel, zas jen testy na atributy.
- Logický program navíc:
  - IDENTITA, tj. test na rovnost dvou termů (hodnot),
  - GENERALIZACE
    - pomocí proměnných
    - pomocí doménové znalosti Background knowledge,
  - REKURZE.

Příklady generalizace

- Parent(Sharon,Bob):-Father(Sharon, Bob).
- Parent(s,b):-Father(s,b).
- Podobně třeba „napadené pole“ ve hrách.
- Doménová znalost:
  - Parent(x,y):-Father(x,y).
  - Parent(x,y):-Mother(x,y).

  Nám pomůže lépe zobecnit následující vztahy:
  - GrandDaughter(A,B):-Mother(A,C),Mother(C,B).
  - GrandDaughter(E,F):-Father(E,G),Mother(G,F).

Doménová znalost

- Známe „rodokmen“ a fakta:
  - GrandDauthter(V,S). GrandDaughter(A,B).

  snažíme se odvodit koncept „vnucka“.

- Známe chemické složení aminokyseliny,
  - vlastnosti typu hasAccid(),
- snažíme se odvodit prostorovou strukturu.

Rekurze

- partita1([]).

  parita1([1,1|s]):-parita1(s).

- Cesta v grafu, předek, tahy ve hře, odvození důkazu atd.

Dva základní přístupy

- FOIL – starší, jednodušší,

- PROGOL - „odzadu“, inverzní rezoluce, „složitější logické předzpracování“.

Příklad

Předpokládejme data:

GrandDaughter(Victor, Sharon) Father(Sharon, Bob) Father(Tom,Bob)

Female(Sharon) Father(Bob,Victor)

Předpokládáme uzavřený svět, tj. negaci všech literálů o GrandDaughter, Father, Female, které tu nejsou uvedeny.

Máme tedy čtyři konstanty–jména.

Pravidlo $GrandDaughter(x, y) \leftarrow$ pokrývá všech 16 možných přiřazení, jedno positivní $GrandDaughter(Victor, Sharon)$ a 15 negativních.

FOIL

- Jeden z mnoha programů na učení logických programů
- velmi podobný učení pravidel postupným pokrýváním
  Sequential covering
- učí Hornovské klauzule se dvěma vyjímkami:
  - nejsou dovoleny funkční symboly v literálech (i tak je prostor k prohledávání dost velký)
  - dovoluje i negované literály v těle, tj. nejen Hornovské klauzule.
- Naučil se rekurzivní definici Quicksortu, rozpoznávat legální a nelegální pozice v šachu.
- Pokrývá jen pozitivní příklady, provádí hill–climbing. ne beam search

Algoritmus **FOIL(Cíl, Predikáty,** *Data***)**

Pos ← positivní příklady v *Data*
Neg ← negativní příklady v *Data*
Pravidla ← {}
while *Pos*, **do** % nauč se nové pravidlo
    NovePravidlo ←pravidlo s prázdným tělem predikující Cíl
    NegPr ← *Neg*
    while *NegPr*, **do** % přidej literál k NovePravidlo
        NoveLiteraly ← k *NovePravidlo* vytvoř kandidáty z *Predikáty* %bude
        Favorit ← *Literál* ∈ *NoveLiteraly*, mající maximální
            FOIL–Gain(*Literál*, *NovePravidlo*, *Data*, Cíl)
        NovePravidlo ← *NovePravidlo* ∪ *Favorit*
        NegPr ← {*k* ∈ *NegPr* splňující tělo *NovePravidlo*}
    Pravidla ← *Pravidla* ∪ {*NovePravidlo*}
    Pos ← *Pos*\ příklady pokryté *NovePravidlo*
return *Pravidla*

Generování kandidátů

Předpokládejme, že tvoříme kandidáty–literály k pravidlu R

$$P(x_1, x_2, \ldots, x_k) \leftarrow L_1, L_2, \ldots, L_n$$

Uvažujeme každý literál L_{n+1} vzniklý jednou z variant:

- $Q(v_1, \ldots, v_r)$, kde Q je predikát z Predikáty, v_i jsou proměnné a aspoň jedna z nich se vyskytuje mezi proměnnými v pravidle (hlavě nebo tělu).
- $Equal(x_j, x_i)$, kde se obě proměnné x_i, x_j vyskytují v pravidle
- negace literálu z předchozích bodů.

Příklad

Učíme predikát $GrandDaughter(x, y)$, další predikáty jsou $Father$ a $Female$.

- První návrh pravidla je $GrandDaughter(x, y) \leftarrow$.
- Zkoušíme přidat každý z literálů:
  $Equal(x, y), Female(x), Female(y), Father(x, y), Father(y, x),$
  $Father(x, z), Father(z, x), Father(y, z), Father(z, y)$ a negaci každého z nich.
- Řekněme, že FOIL vybere $Father(y, z)$ a generuje pravidlo:
  $GrandDaughter(x, y) \leftarrow Father(y, z)$
- Pro další rozšíření uvažuji všechny výše uvedené literály a navíc $Female(z), Equal(z, x), Equal(z, y), Father(z, w), Father(w, z)$ a jejich negace.

- FOIL vybere (např.) $Father(z, x)$ a po ještě jedné iteraci $Female(y)$ a vytvoří pravidlo:
  $GrandDaughter(x, y) \leftarrow Father(y, z) \& Father(z, x) \& Female(y)$
- Toto pravidlo nepokrývá žádný negativní příklad, tedy uzavřeme tvorbu pravidla, přidáme ho do množiny pravidel a odstraníme positivní příklady pokryté tímto pravidlem. Pokud zbyly positivní příklady, učíme další pravidla.

Výběr literálu

FOIL–Gain(*Literál*, *NovePravidlo*, *Data*, Cíl)=$t\left(log_2 \frac{p_1}{p_1+n_1} - log_2 \frac{p_0}{p_0+n_0}\right)$

kde:

p_0	počet positivních přiřazení proměnným v pravidle R (bez nového literálu)
n_0	počet negativních přiřazení proměnným v pravidle R
p_1	počet positivních přiřazení proměnným v pravidle R′ (s novým literálem)
n_1	počet negativních přiřazení proměnným v pravidle R′
t	počet positivních přiřazení pravidlu R, která jsou pokrytá i pravidlem R′

Negativním přiřazením se myslí takové přiřazení, že pro něj v Datech cílový predikát neplatí.

Positivní přiřazení R je pokryto R', pokud aspoň jedno rozšíření o přidané proměnné je přiřazením v R'.

Rozšíření

Rekurzivní pravidla

Pokud do Predikáty přidáme i cílový predikát, můžeme se učit i rekurzivní pravidla. Musíme jen přidat kontrolu nekonečně vnořené rekurze.

FOIL byl schopen se naučit některé rekurzivní predikáty.

Šum

Na datech bez šumu můžeme přidávat literály do té doby, až není pokryt žádný negativní příklad. Na datech s šumem používáme kombinaci přesnosti, pokrytí a složitosti pravidla. FOIL používá princip minimální délky zápisu. FOIL navíc používá prořezávání (post–prunning).

Pohled logiky: Vstupy ILP

- Trénovací příklady E ve formě positivních literálů E^+ a negativních literálů E^-.
- Doménová znalost B ve formě logického programu
  - (požadujeme nutnost = neodvodím vše z E a bezespornost = neodvodím nic z aE^-).
- Chceme hypotézu H ve formě logického programu, pro kterou:
  - $B,H \vdash E^+$ (nebo aspoň většina z nich)
  - $B,H \not\vdash E^-$ (nebo aspoň většina z nich).

Inverzní rezoluce

- Pokud známe:
  - GrandDaughter(v,s):-Father(s,b), Father(B,V), Female(s).
  - Father(Sharon,Bob). Father(Bob, Victor). Female(Sharon).
- Můžeme rezolucí odvodit:
  - GrandDaughter(Victor, Sharon).

- Inverzní rezoluce zná výsledek a fakta, snaží se odvodit pravidlo.

Inverzní subsumpce

1. Positivní příklady relativizovat veškerou znalostí.
2. Převést na klauzule.
3. Anti-unifikovat, co lze, tj. nahradit termy proměnnými.
4. Vymazat negativní literály obsahující proměnné, které se nevyskytují v pozitivních literálech

   (resp. opatrněji, jinak nenajdu GrandDaughter).
5. Klauzule převeď zpět do hornovských klauzulí.

Ještě trochu složitější...

- Progol, Imparo – nejdříve konstruují pomocnou bridge teorii F, která je důsledkem
  - B & ¬ E

  a jejímž důsledkem je
  - ¬ H.