Uvod v relacijsko učenje in induktivno logično programiranje

Ljupčo Todorovski

Univerza v Ljubljani, Fakulteta za upravo
Institut Jožef Stefan, Odsek za tehnologije znanja (E8)

April 2019

Primer BIB: bibliografska baza, avtorstva in reference

Avtorstvo	*Avtor*	*Naslov*
	quinlan	*learning logical definitions from relations*
	lloyd	*foundations of lp*
	lloyd	*logic for learning*
	russell	*ai a modern approach*
	norvig	*ai a modern approach*
	muggleton	*ilp theory and methods*
	deraedt	*ilp theory and methods*

Referenca	*Naslov*	*Naslov*
	logic for learning	*foundations of lp*
	logic for learning	*learning logical definitions from relations*
	logic for learning	*ai a modern approach*
	ai a modern approach	*foundations of lp*
	ai a modern approach	*ilp theory and methods*
	ilp theory and methods	*learning logical definitions from relations*
	ilp theory and methods	*foundations of lp*

Primer BIB: ciljna relacija citati

Citat	*Avtor*	*Avtor*
	lloyd	*lloyd*
	lloyd	*quinlan*
	lloyd	*russel*
	russel	*lloyd*
	norvig	*lloyd*
	⋮	

Naloga relacijskega učenja

- Učimo se definicijo ciljne relacije citat
- Pri tem lahko uporabljamo relacije avtorstvo in referenca

Primer BIB: učni primeri in predznanje

Učni primeri

Podatki iz tabele/relacije *Citat*

Predznanje

Podatki iz dveh tabel/relacij *Avtorstvo* in *Referenca*

Razlika od običajnega strojnega učenja

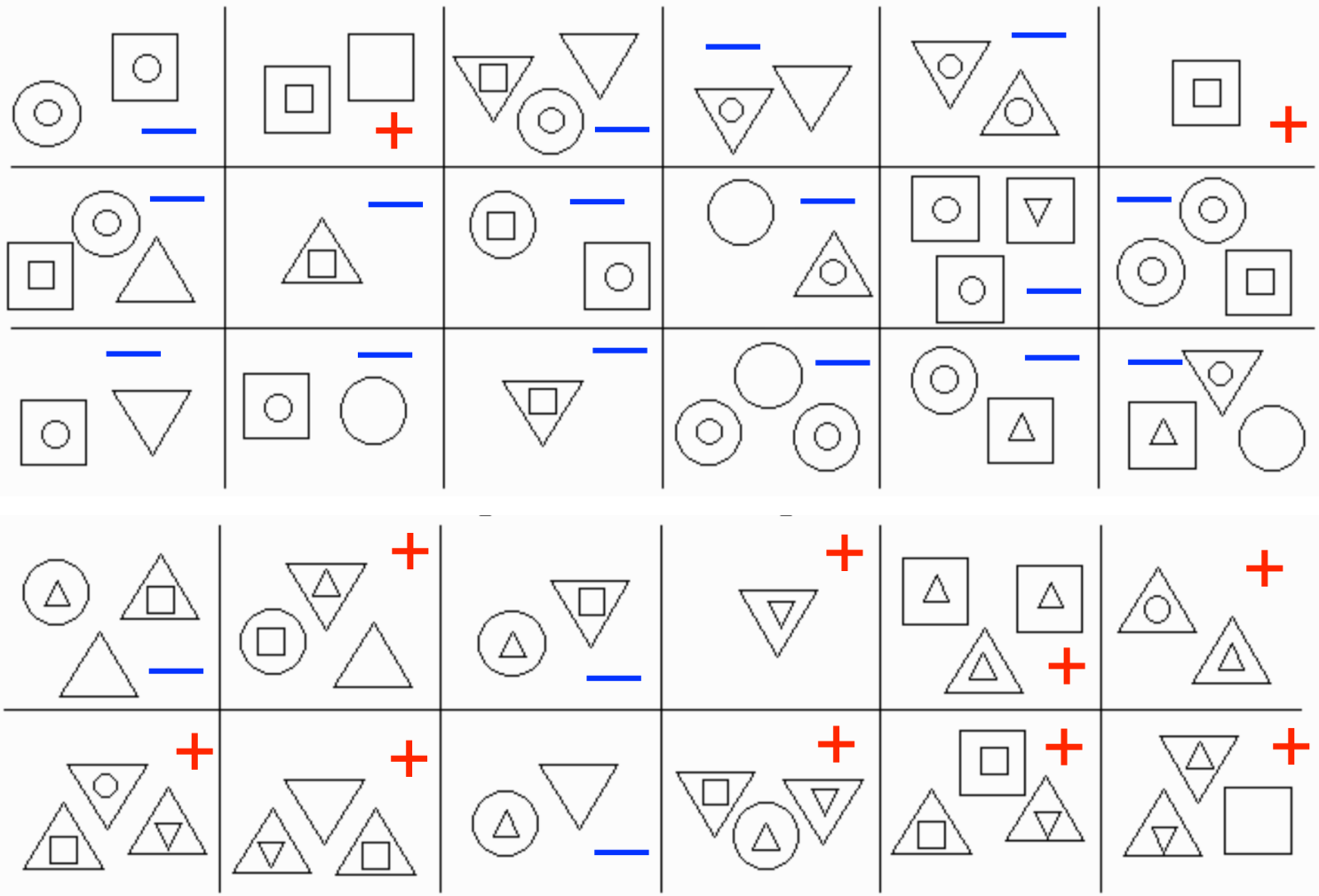
Primeri niso zapisani v eni tabeli.

Primer BIB: rezultat učenja

- ČE
  - je A_1 avtor N_1 in
  - N_1 vsebuje referenco na N_2 in
  - je A_2 avtor N_2
- POTEM A_1 citira A_2

$$Avtorstvo(A_1, N_1) \wedge Referenca(N_1, N_2) \wedge Avtorstvo(A_2, N_2) \Rightarrow Citat(A_1, A_2)$$

Primer BONGARD: primeri



Primer BONGARD: predikatni opis primerov

Prvi negativni primer	krog(k1), krog(k2), krog(k3), kvadrat(v1), vsebuje(k1, k2), vsebuje(v1, k3)
Prvi pozitivni primer	kvadrat(v1), kvadrat(v2), kvadrat(v3), vsebuje(v1, v2)
Tretji pozitivni primer	krog(k1), kvadrat(v1), trikotnik(t1), trikotnik(t2), trikotnik(t3), vsebuje(k1, v1), vsebuje(t1, t2)

Razlika od običajnega strojnega učenja

Primeri

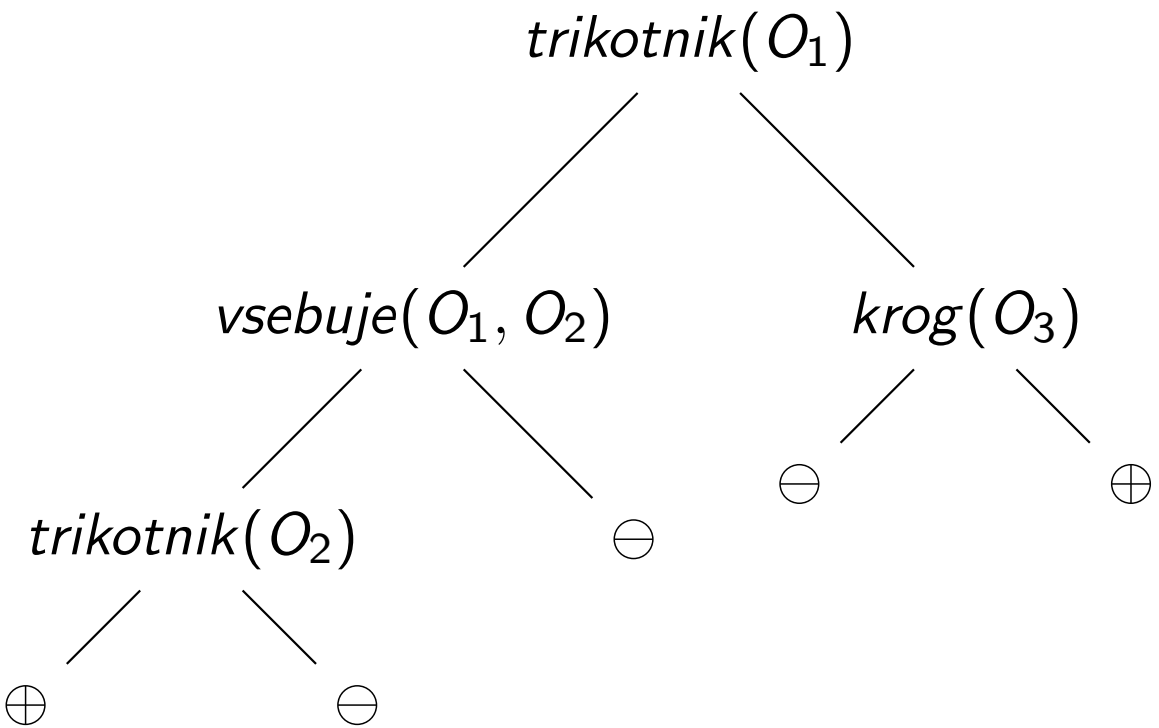
- So množice namesto p-teric
- Vsebujejo strukturirane izraze (predikate)

Model za razvrščanje: Pozitivni primer

- Ima vsaj en trikotnik vsebovan v drugem trikotniku
- Sicer, če nima trikotnika ne sme imeti niti kroga

Kako formalno zapišemo model?

Relacijsko drevo



Kakšna je semantika relacijskega drevesa?

(Ne tako zelo) kratek izlet v logiko prvega reda (stavčno logiko).

Elementi sintakse stavčne logike (*clausal logic*)

- Izrazi (*terms*)
- Atomi (*atoms*)
- Stavki (*clauses*)

Pomembna razlika med izrazi in atomi

- Za izraze ne ugotavljamo njihove resničnosti
- Atomom lahko priredimo resničnostno vrednost

Izrazi: Konstante in sestavljeni izrazi

Se nanašajo na objekte v opazovani domeni.

Konstante: nizi znakov z malo začetnico

Primera: *lloyd* (BIB), *t1* (BONGARD)

Spremenljivke: nizi znakov z veliko začetnico

- Lahko jim priredimo poljubni izraz kot vrednost
- Primeri: *Avtor*, *A*, *Oblika*, *O*

Strukture ali sestavljeni izrazi (*compound terms*)

- Funkcija f/n, kjer n označuje število argumentov f
- Primer *oseba/2: oseba(lloyd, moški)*

Opredeljenost izrazov

Popolnoma opredeljen izraz (*ground term*)

Nanaša se na en objekt iz opazovane domene. Je lahko

- Konstanta
- Sestavljeni izraz brez spremenljivk

Primeri

- *lloyd* in *oseba(lloyd, moški)* sta popolnoma opredeljena izraza
- *Oseba* in *oseba(Oseba, ženska)* nista, saj se lahko nanašata na katerokoli osebo *Oseba* (objekt iz opazovane domene)

Atomi in literali

Predikati

- Predikat p/n, kje n označuje število argumentov p
- Predstavlja relacijo med izrazi
- Primer: *avtorstvo/2: avtorstvo(lloyd, foundations_of_lp)*
- Primer: *trikotnik/1: trikotnik(t1)*

Razlika med atomi (predikati) in (sestavljenimi) izrazi

- Strukturirani izrazi se nanašajo na objekte v domeni
- Atomi (predikati) se nanašajo na relacije med objekti
- Za atome lahko ugotavljamo resničnostno vrednost

Literal je lahko atom ali logična negacija atoma.

Stavki in teorija

Stavke tvorimo iz atomov (literalov) z implikacijo

$$h_1; h_2; \ldots; h_m \leftarrow b_1, b_2, \ldots, b_n$$

- h_j in b_i so atomi oz. literali
- Glava (*head*) stavka je disjunkcija $h = h_1 \vee h_2 \vee \ldots \vee h_m$
- Telo (*body*) stavka je konjunkcija $b = b_1 \wedge b_2 \wedge \ldots \wedge b_n$
- Logični pomen stavka je $b \Rightarrow h$

Če upoštevamo, da je $p \Rightarrow q$ isto kot $\neg p \vee q$

$$\neg b_1 \vee \neg b_2 \vee \ldots \vee \neg b_n \vee h_1 \vee h_2 \vee \ldots \vee h_m$$

Teorija (ali logični program) je množica stavkov.

Primeri stavkov brez spremenljivk

Osebe so lahko srečne ali žalostne.

$$srecna; zalostna \leftarrow oseba.$$

- Oseba, ki ni srečna je žalostna.
- Oseba, ki ni žalostna je srečna.
- Pozor: oseba je lahko hkrati srečna in žalostna.

Ni osebe, ki bi bila srečna in žalostna.

$$\leftarrow oseba, srecna, zalostna.$$

Tudi: Žalostne osebe niso srečne (in obratno).

Primeri stavkov s spremenljivkami

Vsaka miška ima rep.

$$tail\_of(t, X) \leftarrow mouse(X).$$

Ni trikotnika, ki bi vseboval drug trikotnik.

$$\leftarrow trikotnik(X), trikotnik(Y), vsebuje(X, Y).$$

Posebni razredi stavkov

- Dejstva: prazno telo in en atom v glavi, $m = 1$, $n = 0$
- Definitni (določeni) stavki: en atom v glavi, $m = 1$, n poljuben
- Zanikanja ali poizvedbe: prazna glava, $m = 0$, n poljuben
- Hornovi stavki: največ en atom v glavi $m \leq 1$, n poljuben

Primeri posebnih stavkov: korak k semantiki

Dejstvo: $trikotnik(t1) \leftarrow .$

$$\neg\top \vee trikotnik(t1) \equiv \bot \vee trikotnik(t1) \equiv trikotnik(t1)$$

Zanikanje ali poizvedba: $\leftarrow trikotnik(t1).$ in $\leftarrow trikotnik(T).$

$$\neg trikotnik(t1) \vee \bot \equiv \neg trikotnik(t1)$$

$$\neg\forall T : trikotnik(T) \vee \bot \equiv \exists T : \neg trikotnik(T) \vee \bot \equiv \exists T : \neg trikotnik(T)$$

Hierarhija tipov stavčne logike

Propozicijska logika

Izrazi so lahko le resnični ali neresnični: ni spremenljivk.

Relacijska logika

Objekte lahko naslavljamo s konstantami ali spremenljivkami.

Polna logika

Lahko imamo strukturirane izraze.

Definitna (določena) logika

Le defintini (določeni) stavki: ni disjunkcije v glavi stavka.

Zamenjava (*substitution*) in instanca stavka

Zamenjava preslika spremenljivke v izraze

Od neopredeljenih proti popolnoma opredeljenimi izrazi (stavki).

Instanca stavka

- Ko zamenjavo apliciramo na stavek, dobimo njegovo *instanco*
- Če instanca ne vsebuje spremenljivk, dobimo popolnoma določen stavek (popolnoma določeno instanco)

Vse instance stavka so njegove logične posledice.

Primer zamenjave

Stavek c pred zamenjavo

$$tail\_of(t(X), X) \leftarrow mouse(X).$$

Zamenjava θ

$$\theta : X = mickey$$

Instanca stavka c_θ

$$tail\_of(t(mickey), mickey) \leftarrow mouse(mickey).$$

Je popolnoma določena instanca, saj ne vsebuje spremenljivk.

Herbradnove interpretacije

Herbradnov univerzum

Množica vseh določenih izrazov (konstant).

Herbrandova baza

Množica vseh določenih atomov, t.j., literalov brez spremenljivk.

Herbrandova interpretacija

Množica *resničnih* določenih atomov.

Resničnost stavka in Herbrandov model

Resničnost stavka

- Stavek je *neresničen za podano interpretacijo*, če so vsi literali *v telesu resnični* in vsi literali *v glavi neresnični*.
- Sicer je stavek *resničen za podano interpretacijo*

Herbradnov model stavka

Interpretacija v kateri so vse določene instance stavka resnične.

Primer BONGARD

Podana sta

- Herbrandov univerzum *k1*, *v1*, *t1*, *t2*, *t3* ter
- Herbrandova interpretacija *krog(k1)*, *kvadrat(v1)*, *trikotnik(t1)*, *trikotnik(t2)*, *trikotnik(t3)*, *vsebuje(k1, v1)*, *vsebuje(t1, t2)*

Preveri veljavnost stavka (oz. uspešnost poizvedbe)

$$\leftarrow trikotnik(X), trikotnik(Y), vsebuje(X, Y).$$

Poizvedba ima obliko stavka zanikanja zaradi izpeljave s protislovjem (glej naslednjo prosojnico).

Primer BONGARD: Logična izpeljava s protislovjem

$\leftarrow trikotnik(X), trikotnik(Y), vsebuje(X, Y).$

Zamenjava $X = t1$

$\leftarrow trikotnik(t1), trikotnik(Y), vsebuje(t1, Y).$

Zamenjava $Y = t2$

$\leftarrow trikotnik(t1), trikotnik(t2), vsebuje(t1, t2).$

$$\neg trikotnik(t1) \vee \neg trikotnik(t2) \vee \neg vsebuje(t1, t2)$$

Protislovje s Herbradnovo interpretacijo na prejšnji prosojnici.

Stavek torej *ni resničen*, a to vendar pomeni, da je *poizvedba uspešna*.

Semantika rezultata poizvedbe

Uspešna poizvedba

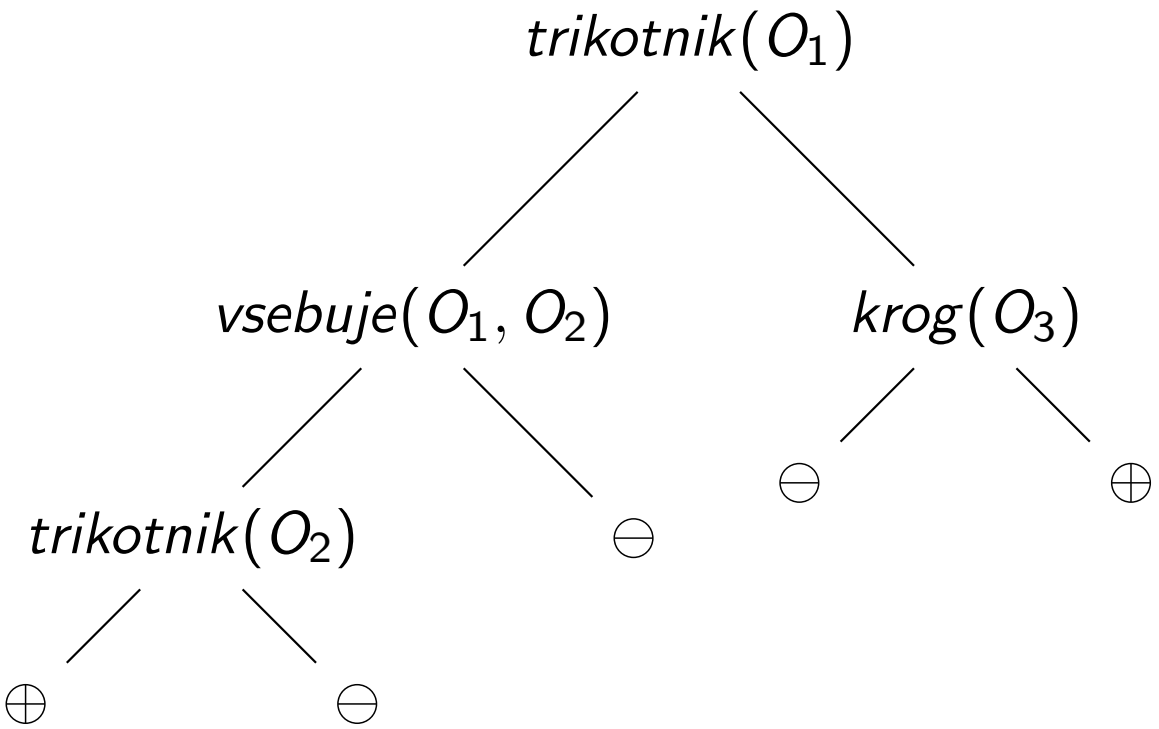
- Ob zamenjavi $\theta = \{X = t1, Y = t2\}$
- In je torej rezultat poizvedbe $X = t1, Y = t2$
- Pozor: rešitev poizvedbe je lahko več!

Pomen uspeha poizvedbe v logiki prvega reda

$$\exists X, Y : \mathit{trikotnik}(X) \wedge \mathit{trikotnik}(Y) \wedge \mathit{vsebuje}(X, Y)$$

Rečemo tudi, da poizvedba *pokriva* podano Herbradnovo interpretacijo.

Relacijska odločitvena drevesa: vozlišča



Pomen vozlišč

- Notranja vozlišča: poizvedbe, npr. $\leftarrow$ *trikotnik*(O_1)
- Končna vozlišča: napovedi

Relacijska odločitvena drevesa: veje

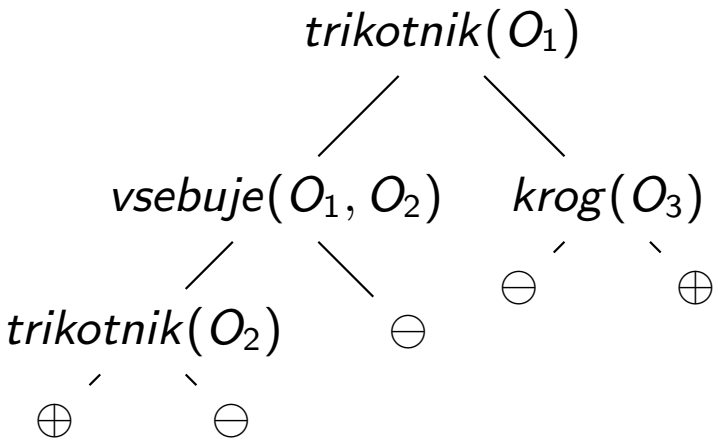
Leva veja

- Poizvedba v vozlišču uspešna
- Spremenljivke dobijo vrednosti iz rešitev poizvedbe
- Vrednosti spremenljivk *veljavne le v levi veji*

Desna veja

- Poizvedba v vozlišču ni uspešna
- Vrednosti spremenljivk iz poizvedbe neznane
- Zato tudi spremenljivke nimajo vrednosti v desni veji

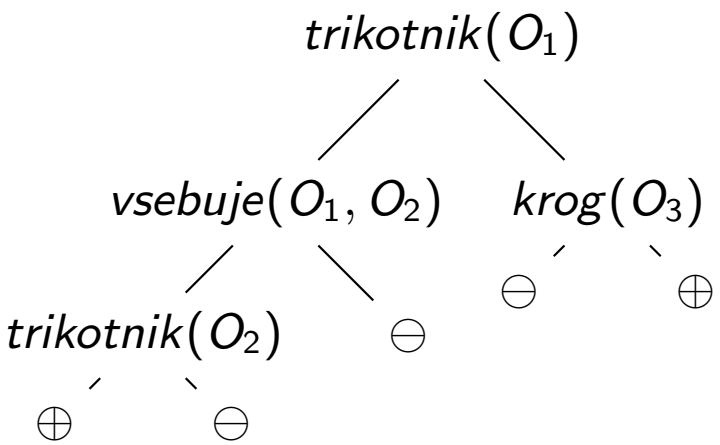
Relacijska odločitvena drevesa: napovedi



Končna vozlišča od leve proti desni

1. $\exists O_1, O_2 : trikotnik(O_1) \wedge vsebuje(O_1, O_2) \wedge trikotnik(O_2) \Rightarrow \oplus$
2. $\exists O_1, O_2 : trikotnik(O_1) \wedge vsebuje(O_1, O_2) \wedge \neg trikotnik(O_2) \Rightarrow \ominus$
3. $\exists O_1 : trikotnik(O_1) \wedge \nexists O_2 : vsebuje(O_1, O_2) \Rightarrow \ominus$
4. $\nexists O_1 : trikotnik(O_1) \wedge \exists O_3 : krog(O_3) \Rightarrow \ominus$
5. $\nexists O_1 : trikotnik(O_1) \wedge \nexists O_3 : krog(O_3) \Rightarrow \oplus$

Relacijska odločitvena drevesa: IF-THEN-ELSE



IF $trikotnik(O_1) \wedge vsebuje(O_1, O_2) \wedge trikotnik(O_2)$ THEN $\oplus$
ELIF $trikotnik(O_1) \wedge vsebuje(O_1, O_2)$ THEN $\ominus$
ELIF $trikotnik(O_1)$ THEN $\ominus$
ELIF $krog(O_3)$ THEN $\ominus$
ELSE $\oplus$

Primer BIB: bibliografska baza, avtorstva in reference

Avtorstvo	*Avtor*	*Naslov*
	quinlan	*learning logical definitions from relations*
	lloyd	*foundations of lp*
	lloyd	*logic for learning*
	russell	*ai a modern approach*
	norvig	*ai a modern approach*
	muggleton	*ilp theory and methods*
	deraedt	*ilp theory and methods*

Referenca	*Naslov*	*Naslov*
	logic for learning	*foundations of lp*
	logic for learning	*learning logical definitions from relations*
	logic for learning	*ai a modern approach*
	ai a modern approach	*foundations of lp*
	ai a modern approach	*ilp theory and methods*
	ilp theory and methods	*learning logical definitions from relations*
	ilp theory and methods	*foundations of lp*

Stavčna logika in podatkovne baze

$\leftarrow$ *Avtorstvo*(*lloyd*, *Naslov*)

```
SELECT Naslov FROM Avtorstvo WHERE Avtor = lloyd
```

Rešitvi *Naslov* = *foundations of lp*, *Naslov* = *logic for learning*.

$\leftarrow$ *Avtorstvo*(*A*, *logic_for_learning*), *Avtorstvo*(*A*, *foundations_of_lp*)

```
SELECT Avtor FROM Avtorstvo t1, Avtorstvo t2 WHERE
       t1.Avtor = t2.Avtor AND t1.Naslov = logic for learning AND
       t2.Naslov = foundations of lp
```

Rešitev *Avtor* = *lloyd*

Primer BIB: rezultat učenja je poizvedba

Citat	*Avtor* A_1	*Avtor* A_2
	lloyd	*lloyd*
	lloyd	*quinlan*
	lloyd	*russel*
	russel	*lloyd*
	norvig	*lloyd*
	$\vdots$	

Rezultat učenja je poizvedba *Citat*

$$\leftarrow Avtorstvo(A_1, N_1), Referenca(N_1, N_2), Avtorstvo(A_2, N_2)$$

Pokritost in pokritje

Pokritost $covers(p, e)$

- p je poizvedba
- e je Herbrandova interpretacija

Pokritje $coverage(p)$

$$coverage(p) = \{e : covers(p, e)\}$$

Množica vseh Herbradnovih interpretacij, za katere je poizvedba uspešna.

Zakaj je to pomembno?

Zato ker nam lahko pomaga strukturirati prostor poizvedb.

Relacija posploševanja

p_1 je *bolj splošna* od poizvedbe p_2, $p_1 \preceq p_2$:

$$coverage(p_2) \subseteq coverage(p_1)$$

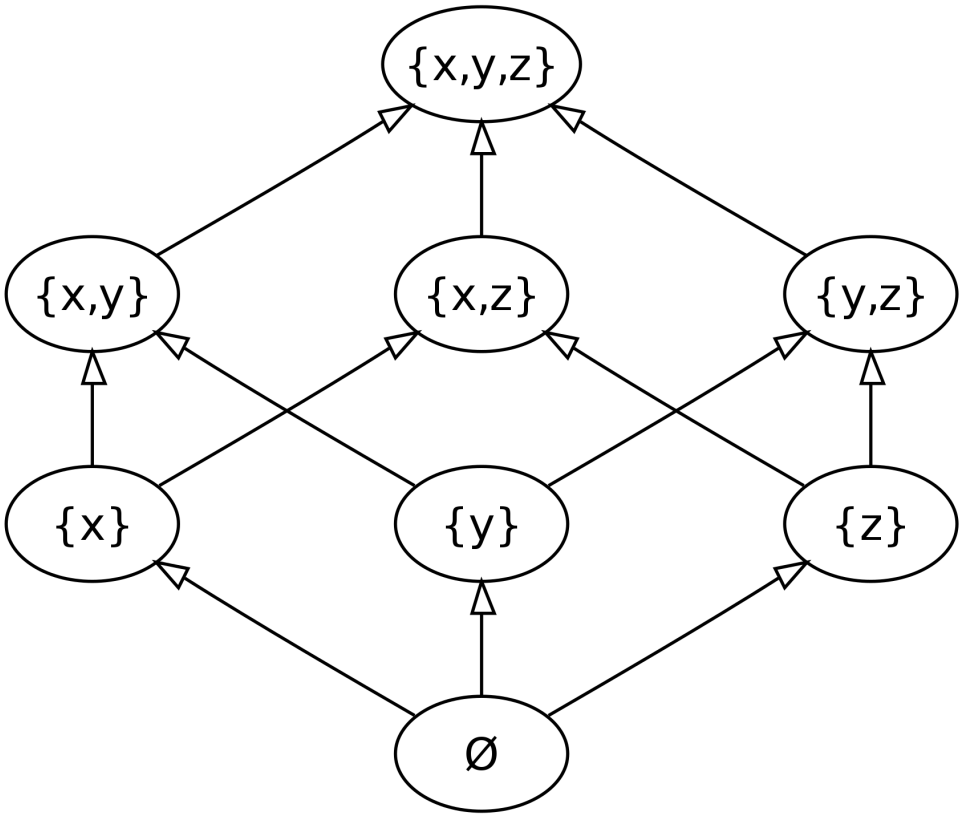
Rečemo tudi, da je p_2 *bolj specifična* od p_1.

Robni poizvedbi

- $\leftarrow \top$: pokriva vse Herbrandove interpretacije
- $\square$: ne pokriva nobene Herbradnove interpretacije, $coverage(\square) = \emptyset$

Hassejev diagram

Relacija posploševanja delno uredi prostor hipotez



Operatorji izostritve in njihove lastnosti

Operator izostritve ρ

$$\rho(p) = \{q : p \preceq q\}$$

Od splošnega k specifičnemu (lahko tudi od specifičnega k splošnemu)

Lastnosti operatorjev izostritve

- Idealnost: nasledniki so najbolj splošne poizvedbe
- Popolnost: iz najbolj splošne poizvedbe z zaporedjem izostritev pridemo do katerokoli druge poizvedbe *vsaj na en način*
- Optimalnost: isto kot zgoraj, a *na točno en način*

θ-subsumpcija (θ-*subsumption*)

p_1 θ-subsuma p_2, $p_1 \preceq_\theta p_2$:

$$\exists\theta : L(p_1|\theta) \subseteq L(p_2)$$

- θ je zamenjava
- $p_1|\theta$ je stavek p_1 po zamenjavi θ
- $L(p)$ je množica literalov iz disjunktivne oblike p

Ustrezen operator izostritve ρ_θ

$$\rho_\theta(p) = \{q : p \preceq_\theta q\}$$

Lastnosti θ-subsumpcije

Enostaven za implementacijo

- Stavku dodamo poljuben literal
- Zamenjavo lahko izvedemo z dodajanjem literalov oblike $X = Y$

Težave s semantiko

- $p \preceq_\theta q \Rightarrow p \preceq q$, ampak $p \preceq q \not\Rightarrow p \preceq_\theta q$
- Zato je možno $p \preceq_\theta q$ in $q \preceq_\theta p$ tudi za $p \neq q$
- Takrat rečemo, da sta p in q sintaktični varianti istega stavka
- Posledica sintaktičnih variant so težave z optimalnostjo izostritve

Generični algoritem od splošnega k specifičnemu

function $G2S(Init, Stop, Cond)$
  $Q = Init$
  $R = \emptyset$
  while not $Stop$ **do**
    izberi p iz Q: $Q = Q \setminus \{p\}$
    if $Cond(p)$ **then**
      $R = R \cup \{p\}$
      $Q = Q \cup \rho(p)$
  return R

Izčrpno iskanje

- $Init = \{\top\}$
- $Stop = (Q = \emptyset)$
- ρ je optimalen operator izostritve

Iskanje optimalnega testa

Algoritem *G2S*

- Pri iskanju prvega testa je $Init = \{\top\}$
- $Cond(p) = (coverage(p) \neq \emptyset)$
- $Stop = (Q \neq \emptyset)$
- Operator izostritve ρ je ρ_θ

Kaj pomeni *čista* množica primerov?

Funkcija nečistoče $Impurity(S)$

Funkcija nečistoče meri varianco vrednosti ciljne spremenljivke Y v S.

Regresija, $D_Y \subseteq \mathbb{R}$

$$Impurity(S) = \frac{1}{|S|} \sum_{(\boldsymbol{x},y)\in S} (y - \bar{y})^2$$

Klasifikacija, $D_Y = \{v_1, v_2, \ldots v_c\}$

$$Impurity(S) = \phi(p_1, p_2, \ldots p_c)$$

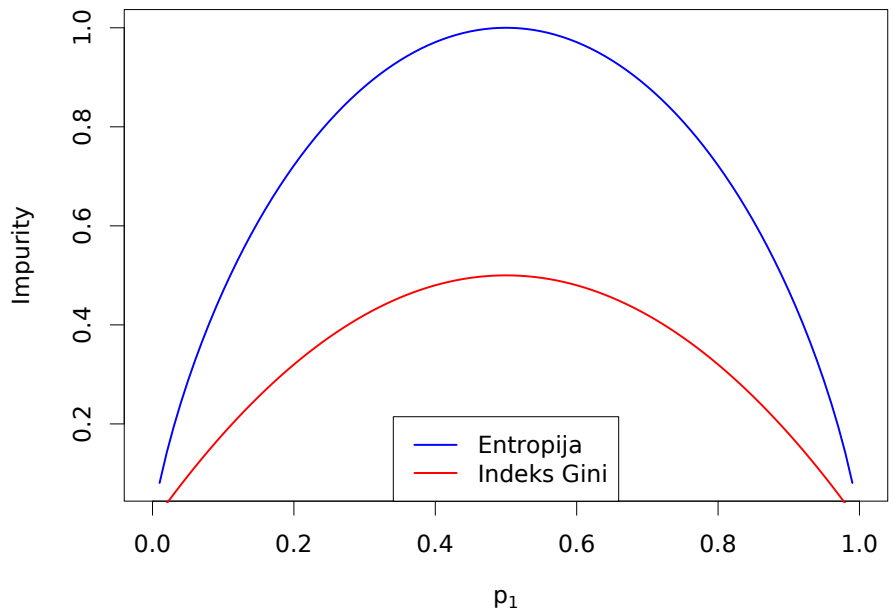
Verjetnosti $p_i = p(Y = v_i|S)$

Zaželene lastnosti funkcije nečistoče $\phi(p_1, p_2, \ldots p_c)$

- Doseže maksimalno vrednost pri enakomerni porazdelitvi $\forall i : p_i = 1/c$
- Doseže minimalno vrednost pri porazdelitvah, kjer $\exists! i : p_i = 1$
- Simetrična: neobčutljiva na vrstni red parametrov
- Konkavna, zvezna in zvezno odvedljiva

Dve pogosto uporabljani funkciji

1. Entropija $\phi(p_1, p_2, \ldots p_c) = -\sum_{i=1}^{c} p_i \log_2 p_i$
2. Indeks Gini $\phi(p_1, p_2, \ldots p_c) = 1 - \sum_{i=1}^{c} p_i^2$



Izbira optimalnega testa $Split = SelectOptimal(S)$

Ciljna funkcija za optimizacijo je zmanjšanje nečistoče *IR*

$$IR(Split, S) = Impurity(S) - \sum_{i=1}^{s} \frac{|S_i|}{|S|} Impurity(S_i)$$

- Test *Split* povzroči razbitje S na $\{S_1, S_2, \ldots S_s\}$
- $IR = Impurity\ Reduction$

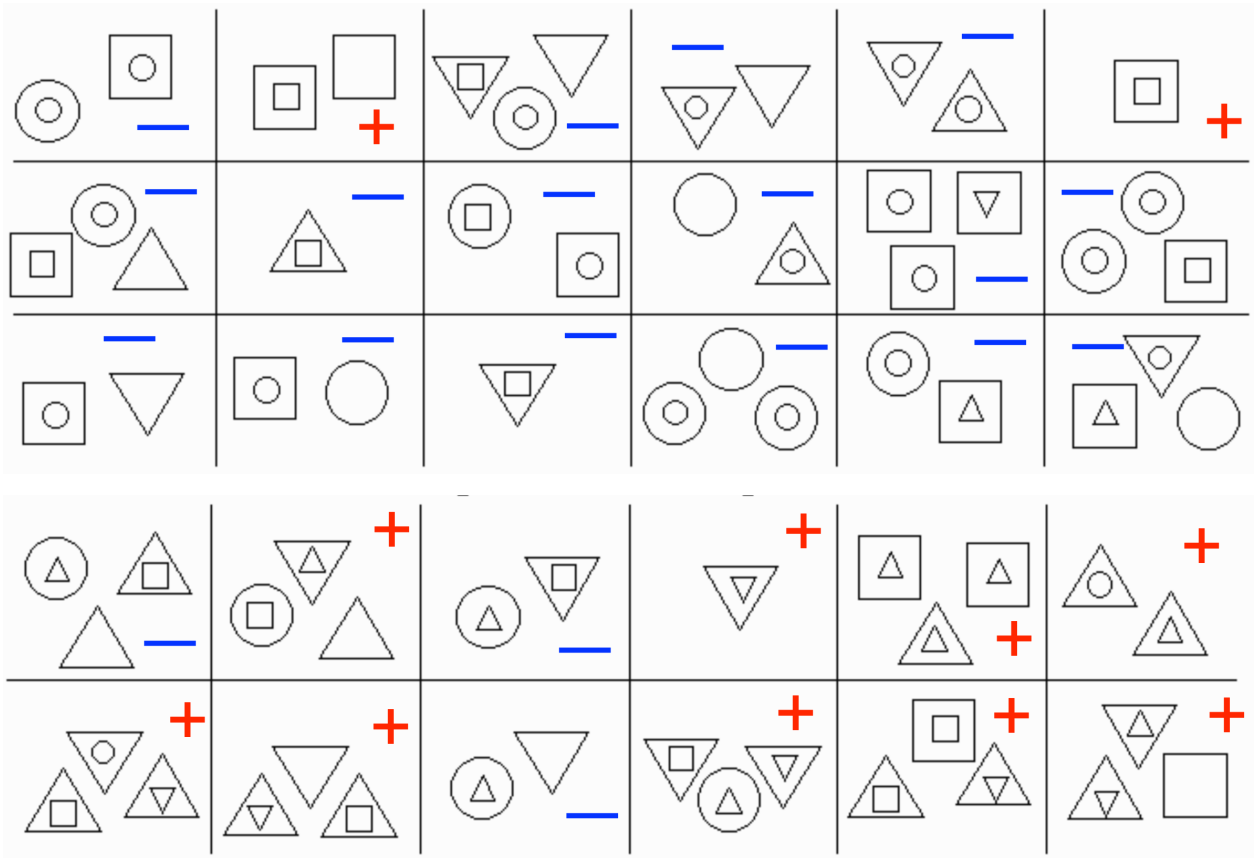
Test izberemo z optimizacijo

$$\max_{Split} IR(Split, S) = \min_{Split} \sum_{i=1}^{s} \frac{|S_i|}{|S|} Impurity(S_i)$$

Možne teste *Split* naštejemo z algoritmom *G2S*

BONGARD: 30 primerov za dvojiško razvrščanje

Primere oštevilčimo od leve proti desni, nato navzdol



BONGARD: Herbrandove interpretacije primerov

$e_1, \ominus$	krog(k1), krog(k2), krog(k3), kvadrat(v1), vsebuje(k1, k2), vsebuje(v1, k3)
$e_2, \oplus$	kvadrat(v1), kvadrat(v2), kvadrat(v3), vsebuje(v1, v2)
$e_{20}, \oplus$	krog(k1), kvadrat(v1), trikotnik(t1), trikotnik(t2), trikotnik(t3), vsebuje(k1, v1), vsebuje(t1, t2)

BONGARD: Nečistost učne množice in testi

$Impurity(S), S = \{e_1, e_2, \ldots, e_{30}\}$

- $p(\oplus) = 11/30$, $p(\ominus) = 19/30$
- $Gini(S) = 1 - ((11/30)^2 + (19/30)^2) \doteq 0.46$

Možne poizvedbe v korenskem vozlišču

1. $\leftarrow krog(O_1)$
2. $\leftarrow kvadrat(O_1)$
3. $\leftarrow trikotnik(O_1)$
4. $\leftarrow vsebuje(O_1, O_2)$

BONGARD: Test $trikotnik(O_1)$

Pozor: $trikotnik(O_1) \equiv \exists O_1 : trikotnik(O_1)$

$Impurity(S_1), |S_1| = 23$, vzorci s trikotnikom

- $p(\oplus) = 9/23,\ p(\ominus) = 14/23$
- $Gini(S_1) = 1 - ((9/23)^2 + (14/23)^2) \doteq 0.48$

$Impurity(S_2), |S_2| = 7$, vzorci brez trikotnika

- $p(\oplus) = 2/7,\ p(\ominus) = 5/7$
- $Gini(S_2) = 1 - ((2/7)^2 + (5/7)^2) \doteq 0.41$

$$IR(trikotnik(O_1)) \doteq 0.46 - (\frac{23}{30}0.48 + \frac{7}{30}0.41) \doteq 0.004$$

Izbor nadaljnjih testov

V levi veji

- Poizvedba $\leftarrow$ *trikotnik*(O_1) je uspela
- V vseh podrejenih vozliščih velja torej $\exists O_1$: *trikotnik*(O_1)
- O_1 lahko zavzame eno od vrednosti, ki ustreza enemu izmed trikotnikov v vzorcu

V desni veji

- Poizvedba $\leftarrow$ *trikotnik*(O_1) ni uspela
- V vseh podrejenih vozliščih velja torej $\nexists O_1$: *trikotnik*(O_1)
- O_1 nima smiselne vrednosti

BONGARD: Test $trikotnik(O_1), vsebuje(O_1, O_2)$

$Impurity(S_1), |S_1| = 18$, vzorci s trikotnikom z vsebino

- $p(\oplus) = 9/18$, $p(\ominus) = 9/18$
- $Gini(S_1) = 1 - ((9/18)^2 + (9/18)^2) = 0.5$

$Impurity(S_2), |S_2| = 5$, vzorci s praznim trikotnikom

- $p(\oplus) = 0/5$, $p(\ominus) = 5/5$
- $Gini(S_2) = 1 - ((0/5)^2 + (5/5)^2) = 0$
- *Čisto vozlišče*, ki napoveduje $\ominus$

$$IR(trikotnik(O_1), vsebuje(O_1, O_2)) \doteq 0.48 - (\frac{18}{23}0.5 + \frac{5}{23}0) \doteq 0.089$$

BONGARD: $trikotnik(O_1), vsebuje(O_1, O_2), trikotnik(O_2)$

Impurity$(S_1), |S_1| = 9$, vzorci s trikotnikom, ki vsebuje trikotnik

- $p(\oplus) = 9/9$, $p(\ominus) = 0/9$
- $Gini(S_1) = 1 - ((9/9)^2 + (0/9)^2) = 0$
- *Čisto vozlišče*, ki napoveduje $\oplus$

Impurity$(S_2), |S_2| = 9$, vzorci s trikotniki, ki vsebujejo le kvadrate ali kroge

- $p(\oplus) = 0/9$, $p(\ominus) = 9/9$
- $Gini(S_2) = 1 - ((0/9)^2 + (9/9)^2) = 0$
- *Čisto vozlišče*, ki napoveduje $\ominus$

$$IR(trikotnik(O_1), vsebuje(O_1, O_2), trikotnik(O_2)) = 0.5 - (\frac{9}{18}0 + \frac{9}{18}0) = 0.5$$

BONGARD: Test $\neg trikotnik(O_1), krog(O_3)$

Pozor: $\neg trikotnik(O_1) \equiv \nexists O_1 : trikotnik(O_1)$

Impurity$(S_1), |S_1| = 5$, vzorci brez trikotnika in s krogom

- $p(\oplus) = 0/5,\ p(\ominus) = 5/5$
- $Gini(S_1) = 1 - ((0/5)^2 + (5/5)^2) = 0$
- *Čisto vozlišče*, ki napoveduje $\ominus$

Impurity$(S_2), |S_2| = 2$, vzorci brez trikotnika in brez kroga

- $p(\oplus) = 2/2,\ p(\ominus) = 0/2$
- $Gini(S_2) = 1 - ((2/2)^2 + (0/2)^2) = 0$
- *Čisto vozlišče*, ki napoveduje $\oplus$

$$IR(\neg trikotnik(O_1), krog(O_3)) \doteq 0.41 - (\frac{5}{7}0 + \frac{2}{7}0) = 0.41$$

Algoritem za učenje relacijskih dreves

Učenje relacijskih dreves iz množice Herbrandovih interpretacij S, pri začetnem klicu je $Q = \top$

function $TDIRDT(S, Q)$
  $\leftarrow Q_b = SelectOptimal(\rho_\theta(\leftarrow Q), S)$
  if $Stop(\leftarrow Q_b, S)$ **then**
    return $leaf(S)$
  $Split = Q_b - Q$
  $S_1 = \{e \in S : covers(\leftarrow Q_b, e)\}$
  $S_2 = S \setminus S_1$
  return $tree(Split, TDIRDT(S_1, Q_b), TDIRDT(S_2, Q))$

Tehnična podrobnost

Specifikacije *rmode*

Določajo obliko literalov, ki jih lahko ρ dodaja poizvedbam

Primer *rmode*$(3 : \mathit{vsebuje}(+O_1, O_2)$ določa

- Poizvedba lahko vsebuje največ 3 literale $\mathit{vsebuje}(O_1, O_2)$
- Znak $+$ pred O_1 določa, da mora imeti O_1 že določene vrednosti, torej mora se pojaviti v enem od prejšnjih literalov poizvedbe
- Obratno znak $-$ bi določal, da se spremenljivka ne sme pojaviti prej
- Oboje lahko zelo pospeši iskanje optimalnega testa

Viri in implementacije

Viri

- Učbenik (De Raedt 2008): Logical and Relational Learning
- Doktorska disertacija (Blockeel 1998)

Implementacije

- Veliko implementacij zahteva Prolog, npr. Alpeh
  `www.cs.ox.ac.uk/activities/programinduction/Aleph/aleph.html`
- Razen nepreverjene
  `starling.utdallas.edu/software/boostsrl/`