Rule Learning

- propositional rules
- FOL rules, ILP
  - logical specification
  - specific ⇒ general algos, inverse resolution (GOLEM, Progol)
  - general ⇒ specific algos (FOIL)
  - extra-logical constraints
  - applications
- learning control rules

Learning First Order Rules

Why?

- geometric knowledge often relational
  $on_top(b3, b1, b2), left_of(b3, b1)$..
  → finite element MESH design
  → floor planning in VLSI design
- temporal knowledge relational
  → sensor data of robots
- relational data bases
  $employee(em1, smith, 2400DM, 8years..), works_in(em1, dept1), uses(dept1, item5)$
  → data mining
- encodings of FOL in feature-value lead to very complex learning tasks
- learn recursive definitions
  $Ancestor(x, y) \leftarrow Parent(x, y)$
  $Ancestor(x, y) \leftarrow Parent(x, z) \wedge Ancestor(z, y)$

Example ILP

Data/Evidence/Examples
Minor-violation(event1)
minor-violation(event2)
minor-violation(event(3)
...

Background Knowledge
car-parked(event1,place1)
bus-lane(place1)
car-parked(event2,place2)
second-row(place2)
corrosion(car1) ...
no-parking(X) ← bus-lane(X)
no-parking(X) ← second-row(X)
...

Rules to learn
minor-violation(X) ← car-parked(XX,Y), no-parking(Y)
minor-violation(X) ← involved-vehicle(X,Y), corrosion(Y)

Logical Specification

ILP combines induction with background knowledge and FOL representation

Conditions for construction of H

Prior Necessity: $B \not\models E^+$

(Prior consistency: $B \wedge E^- \not\models \bot$)

Posterior Sufficiency: $B \wedge H \models E^+$

weak consistency: $B \wedge H \not\models \bot$

Posterior consistency: $B \wedge H \wedge E^- \not\models \bot$

Example

several negative consequences

$$B = \begin{cases} haswings(X) \leftarrow bird(X) \\ bird(X) \leftarrow vulture(X) \end{cases}$$

$E^+ = haswings(tweety) \leftarrow$

ground consequences of $B \wedge \overline{E^+}$:

$$C = \overline{bird(tweety)} \wedge \overline{vulture(tweety)} \wedge \overline{haswings(tweety)}$$

leads to 3 most specific starting clauses

$$H = \begin{cases} bird(tweety) \leftarrow \\ vulture(tweety) \leftarrow \\ haswings(tweety) \leftarrow \end{cases}$$

Employing Posterior Sufficiency

to learn $H = b'_1 \wedge b'_2 \ldots \rightarrow h'$

Sufficiency*: $B \wedge \overline{E^+} \models \overline{H}$

1. CASE: Single example clause, single hyp clause:

$E^+ = h \vee \overline{b_1} \vee \ldots$

$H = h' \vee \overline{b'_1} \vee \ldots$

$\Rightarrow B \wedge (\overline{h \vee \overline{b_1} \vee \ldots}) \models (\overline{h' \vee \overline{b'_1} \vee \ldots})$

$B \wedge \overline{h} \wedge b_1 \ldots \models \overline{h'} \wedge b'_1 \ldots$

Employing Posterior Sufficiency

2. CASE: Multiple examples, single hyp clause:

- $E^+ = e_1 \wedge e_2 \wedge \ldots$ ground atoms,
  C unit consequences of $B \wedge \overline{E^+}$
- $\Rightarrow B \wedge \overline{E^+} \models \overline{E^+} \wedge C$
  $\Rightarrow B \wedge \overline{E^+} \models (\overline{e_1} \vee \ldots) \wedge C$ $B \wedge \overline{E^+} \models (\overline{e_1} \wedge C) \vee \ldots$
  $\Rightarrow H = (e_1 \vee \overline{C}) \wedge \ldots$ set of clauses
- Single clause $\Leftarrow$ most specific clause subsuming all (general to specific search)

Example:

$$B = \begin{cases} father(harry, john) \leftarrow \\ father(john, fred) \leftarrow \\ uncle(harry, jill) \leftarrow \end{cases}$$

$$E^+ = \begin{cases} parent(harry, john) \leftarrow \\ parent(john, fred) \leftarrow \end{cases}$$

$C = father(harry, john) \wedge$
. $father(john, fred) \wedge$
. $uncle(harry, jill)$

most specific clauses:

$e_1 \vee \overline{C}$ = $parent(harry, john) \leftarrow$
$father(harry, john)$,
$father(john, fred)$,
$uncle(harry, jill)$

$e_2 \vee \overline{C}$ = $parent(john, fred) \leftarrow$
$father(harrry, john)$,
$father(john, fred)$,
$uncle(harry, jill)$

Predicate Invention

observational vocabulary: $\mathcal{O} = V(E^+ \cup E^-)$
theoretical vocabul;ary: $\mathcal{T} = V(B) - \mathcal{O}$
invented vocabulary: $\mathcal{I} = V(H) - (\mathcal{T} \cup \mathcal{O})$

CASE: Single example, multiple clause hyp:

AndIntroduction: $X \leftrightarrow (X \wedge Y) \vee (X \wedge \overline{Y})$

$B \wedge \overline{E^+} \models (p \wedge C) \vee (\overline{p} \wedge C)$

H is any set entailing $(\overline{p} \vee \overline{C}) \wedge (p \vee \overline{C})$

OrIntroduction: $X \leftrightarrow (X \vee Y) \wedge (X \vee \overline{Y})$

least general generalization: $lgg(e_1 \vee \overline{C}, e_2 \vee \overline{C})$ = $parent(A, B) \leftarrow$
$father(A, B)$,
$father(C, D)$

$B = min(X, [X]) \leftarrow$

$E^+ = min(2, [3, 2]) \leftarrow$

ground consequences of $B \wedge \overline{E^+}$:
$\overline{min(2, [3, 2])} \wedge min(3, [3]) \wedge min(3, [3])$

most specific starting clauses for predicate invention

$$H = \begin{cases} min(2, [3, 2]) \leftarrow \\ min(2, [2]), min(3, [3]), p(2, 3, [2], [3], [3, 2]) \\ \\ p(2, 3, [2], [3], [3, 2]); min(2, [3, 2]) \\ \leftarrow min(2, [2]), min(3, [3]) \end{cases}$$

$$H' = \begin{cases} min(X, [Y, Z]) \leftarrow \\ min(X, Z), p(X, Y) \\ \\ p(2, 3) \leftarrow \end{cases}$$

General ILP Algo

QH := `Initialize`
repeat

`Delete` H from QH

`Choose` rules $r_1..r_k \in R$ to be applied to H

Apply $r_1..r_k$ to H yields $H_1...H_n$

Add $H_1..H_n$ to QH

`Prune` QH

until `stop-criterion`(QH)

Generalization/ Specialization

$G \models S$

deductive rule $G \longrightarrow S$ 'specialization rule'

inductive rule $G \longleftarrow S$ 'generalization rule'

$\Rightarrow$ Basis for pruning the search space:

- when $B \wedge H \not\models e^+$, no specialization of H will imply e
- when $B \wedge H \wedge e^- \models \bot$, all generalizations of H will be inconsistent too.

Example-driven choice and closed world assumption

Example

Target concept: $GrandDaughter(x, y)$, Predicates: $Father, Female$

start: $Grandaughter(x, y) \leftarrow$

spec. candidates: $x = y, Female(x), Female(y)$
$Father(x, y), Father(y, x), Father(x, z)$
$Father(z, x), Father(y, z), Father(z, y)$
all negations

Choice: $Father(y, z)$

resulting rule: $GrandDaugther(x, y) \leftarrow Father(y, z)$

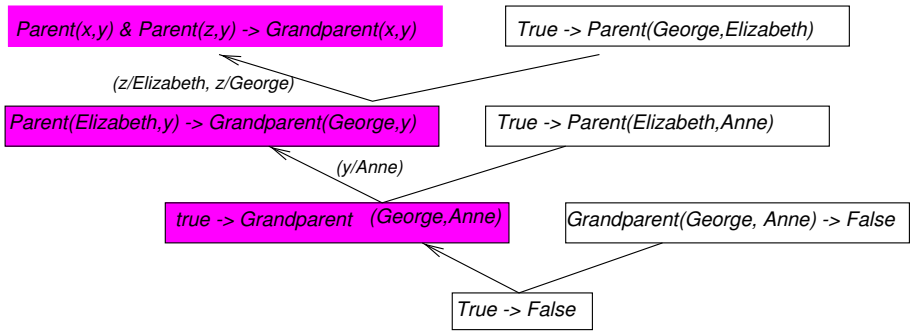
spec. candidates: ... $x = z, z = y$
$Female(z), Father(z, w), Father(w, z)$..

Choice: $Father(z, x)$

Choice: $Female(y)$ covers all pos. examples

$H = GrandDaugther(x, y) \leftarrow Father(y, z) \wedge Father(z, x) \wedge Female(y)$

Inverse Resolution



Inverting resolution

Inductive inference rules:

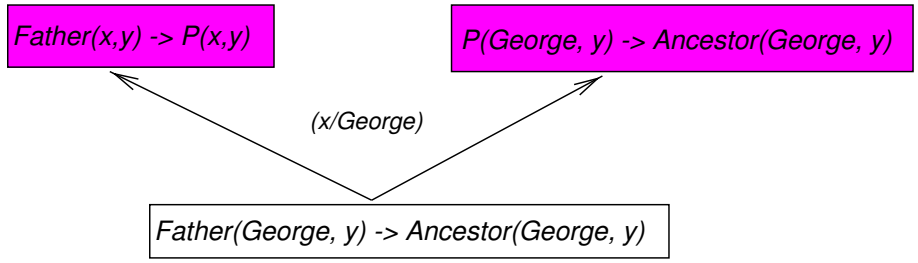
Absorption

Identification

Intra-construction

Inter-construction

Predicate Invention



Subsumption

Definition: $c_1\theta$-subsumes c_2 = exists substitution such that $c_1\theta \subseteq c_2$

h_1 `more general` h_2 special case of θ-`subsumption` special case of `entailment`

inductive rule: $\frac{c_2}{c_1}$, where $c_1\theta \subseteq c_2$

specialization operators under θ-subsumption:
apply substitution, add (set of) literals

Applications: Drug Design

Pharmacophore Discovery: Learn rules for 3-D concepts (ACE inhibitors)

```
Molecule A is an ACE inhibitor if:
   A can bind to zinc at a site B, and
   A contains a hydrogen acceptor C, and
   the distance between B and C is 7.9 +/- 1.0 Angstroms, and
   A contains a hydrogen acceptor D, and
   the distance between B and D is 8.5 +/- Angstroms, and
   the distance between C and D is 2.1 +/- 1.0 Angstrom, and
   A contains a hydrogen acceptor E, and
   ...
```


Applications

Learning chemical bounds fragment in a mass spectrometer (1980)

learning which chemical substructures produce mutagenetic activity (1994)

learning to design finite element meshes (1992)

learning pharmacophore for drug design (1997)