Inductive Logic Programming
Lecture 1.3
Meta-Interpretive Learning of Higher-Order Dyadic Datalog

Stephen Muggleton
Department of Computing
Imperial College, London

5th July, 2018

Paper for this lecture

Paper04: S.H. Muggleton, D. Lin, and A. Tamaddoni-Nezhad. Meta-interpretive learning of higher-order dyadic datalog: Predicate invention revisited. Machine Learning, 2015.

Motivation - revisited

Logic Program [Kowalski, 1980]

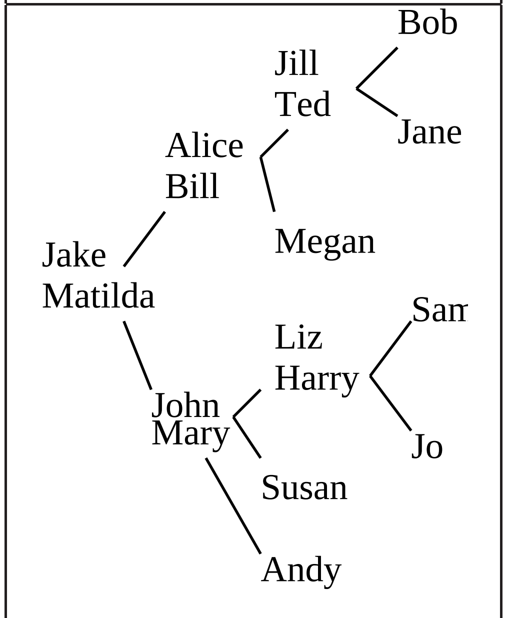
Inductive Logic Programming [Muggleton, 1991]

Machine Learn arbitrary programs

State-of-the-art ILP systems lack Predicate Invention and Recursion [Muggleton et al, 2011]

Family relations (Dyadic)

Family tree



Target Theory

$$father(ted, bob) \leftarrow$$

$$father(ted, jane) \leftarrow$$

$$parent(X, Y) \leftarrow mother(X, Y)$$

$$parent(X, Y) \leftarrow father(X, Y)$$

$$ancestor(X, Y) \leftarrow parent(X, Y)$$

$$ancestor(X, Y) \leftarrow parent(X, Z), \quad ancestor(Z, Y)$$

Meta-Interpretive Learning (MIL)

First-order	Metalogical substitutions
Examples $ancestor(jake, bob) \leftarrow$ $ancestor(alice, jane) \leftarrow$	N/A
Background Knowledge $father(jake, alice) \leftarrow$ $mother(alice, ted) \leftarrow$	N/A
Instantiated Hypothesis $father(ted, bob) \leftarrow$ $father(ted, jane) \leftarrow$ $p1(X, Y) \leftarrow father(X, Y)$ $p1(X, Y) \leftarrow mother(X, Y)$ $ancestor(X, Y) \leftarrow p1(X, Y)$ $ancestor(X, Y) \leftarrow p1(X, Z),$ $ancestor(Z, Y)$	 $metasub(instance, [father, ted, bob])$ $metasub(instance, [father, ted, jane])$ $metasub(base, [p1, father])$ $metasub(base, [p1, mother])$ $metasub(base, [ancestor, p1])$ $metasub(tailrec, [ancestor, p1, ancestor])$

Meta-interpreter

Generalised meta-interpreter
$prove([], Prog, Prog).$
$prove([Atom\|As], Prog1, Prog2) :-$
$\quad metarule(Name, MetaSub, (Atom \text{ :- } Body), Order),$
$\quad Order,$
$\quad save\_subst(metasub(Name, MetaSub), Prog1, Prog3),$
$\quad prove(Body, Prog3, Prog4),$
$\quad prove(As, Prog4, Prog2).$

Metarules

Name	**Meta-Rule**	**Order**
Instance	$P(X,Y) \leftarrow$	$True$
Base	$P(x,y) \leftarrow Q(x,y)$	$P \succ Q$
Chain	$P(x,y) \leftarrow Q(x,z), R(z,y)$	$P \succ Q, P \succ R$
TailRec	$P(x,y) \leftarrow Q(x,z), P(z,y)$	$P \succ Q,$ $x \succ z \succ y$

Logical form of Meta-rules

General form

$$\begin{aligned} P(x,y) &\leftarrow Q(x,y) \\ P(x,y) &\leftarrow Q(x,z), R(z,y) \end{aligned}$$

Meta-rule general form is

$$\exists P, Q, ..\forall x, y, ..P(x, ..) \leftarrow Q(y, ..), ..$$

Supports predicate/object invention and recursion.

In Family Relations we consider datalog logic programs in H^2_2, which contain predicates with arity at most 2 and has at most 2 atoms in the body.

Expressivity of H_2^2

Given an infinite signature H_2^2 has Universal Turing Machine expressivity [Tarnlund, 1977].

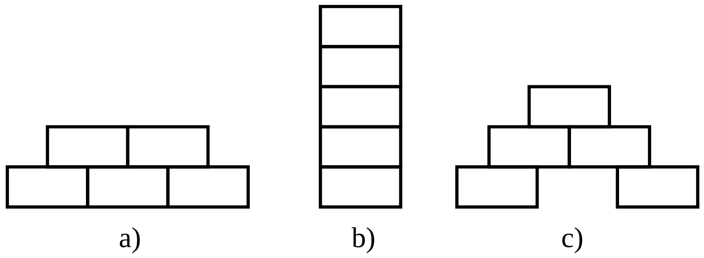
utm(S,S)	$\leftarrow$	halt(S).
utm(S,T)	$\leftarrow$	execute(S,S1), utm(S1,T).
execute(S,T)	$\leftarrow$	instruction(S,F), F(S,T).

Q: How can we limit H_2^2 to avoid the halting problem?

Metagol$_D$ implementation

- Ordered Herbrand Base [Knuth and Bendix, 1970; Yahya, Fernandez and Minker, 1994] - guarantees termination of derivations. Lexicographic + interval.
- Episodes - sequence of related learned concepts.
- $0, 1, 2, ..$ clause hypothesis classes tested progressively.
- Log-bounding (PAC result) - $log_2 n$ clause definition needs n examples.
- Github implementation - https://github.com/metagol/metagol
  .
- PHP interface - http://metagol.doc.ic.ac.uk
  .

Experiment - Robotic strategy learning



Examples of a) stable wall, b) column and c) non-stable wall.

buildWall(X,Y) $\leftarrow$ **a2(X,Y)**, f1(Y)

buildWall(X,Y) $\leftarrow$ **a2(X,Z)**, **buildWall(Z,Y)**

a2(X,Y) $\leftarrow$ **a1(X,Y)**, f1(Y)

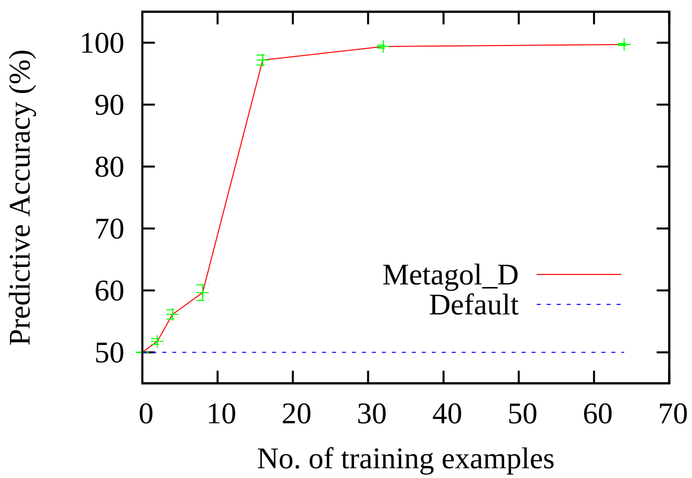
a1(X,Y) $\leftarrow$ **fetch(X,Z)**, **putOnTopOf(Z,Y)**

f1(X) $\leftarrow$ offset(X), continuous(X)

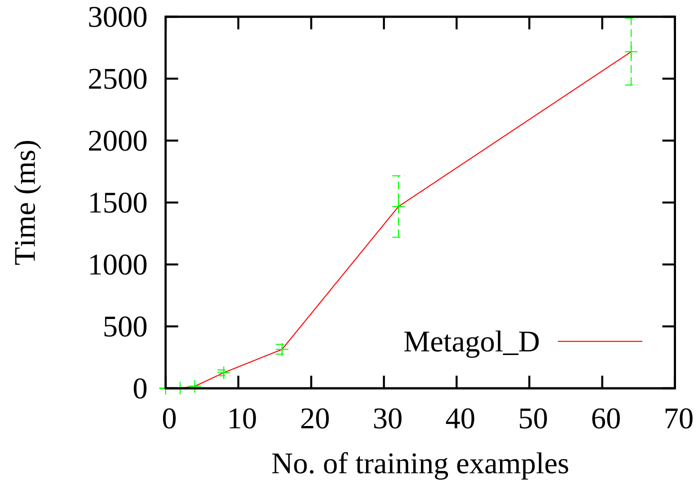
Stable wall strategy built from positive and negative examples. a1, a2 and f1 invented. Dyadic **Actions**, Monadic Fluents.

Performance graphs - Robotic strategy learning

a) Predictive accuracy



b) Learning time



NELL experiment

- CMU's Never Ending Language Learning (NELL), [Carlson et al 2010].
- 50 million facts (triples) from web pages since 2010.

```
playssport(eva_longoria,baseball)
playssport(pudge_rodriguez,baseball)
athletehomestadium(chris_pronger,honda_center)
athletehomestadium(peter_forsberg,wachovia_center)
athletealsoknownas(cleveland_browns,buffalo_bills)
athletealsoknownas(buffalo_bills,cleveland_browns)
```

Metagol$_D$ hypothesis

$$\text{athletehomestadium(X,Y)} \leftarrow \text{athleteplaysforteam(X,Z)},\ \text{teamhomestadium(Z,Y)}$$

Abduced facts

1. athleteplaysforteam(john_salmons,los_angeles_lakers)
2. athleteplaysforteam(trevor_ariza,los_angeles_lakers)
3. athleteplaysforteam(shareef_abdur_rahim,los_angeles_lakers)
4. athleteplaysforteam(armando_marsans,cincinnati)
5. teamhomestadium(carolina_hurricanes,rbc_center)
6. teamhomestadium(anaheim_angels,angel_stadium_of_anaheim)

Abductive hypotheses 2,4,5 and 6 were confirmed using internet search queries. However, 1 and 3 are wrong.

Learning higher-order concepts

Higher-order MetaRule

$$P(X,Y) \leftarrow \mathrm{symmetric}(P),\ P(Y,X)$$

Abduced facts

symmetric(athletealsoknownas) ←

athletealsoknownas(buffalo_bills,broncos) ←

athletealsoknownas(buffalo_bills,kansas_city_chiefs) ←

athletealsoknownas(buffalo_bills,cleveland_browns) ←

Related work

Predicate Invention. Early ILP [Muggleton and Buntine, 1988; Rouveirol and Puget, 1989; Stahl 1992]

Abductive Predicate Invention. Propositional Meta-level abduction [Inoue et al., 2010]

Meta-Interpretive Learning. Learning regular and context-free grammars [Muggleton et al, 2013]

Higher-order Logic Learning. Without background knowledge [Feng and Muggleton, 1992; Lloyd 2003]

Higher-order Datalog. HO-Progol learning [Pahlavi and Muggleton, 2012]

Summary and limitations

Summary

- New form of Declarative Machine Learning [De Raedt, 2012]
- H^2_2 is tractable and Turing-complete fragment of High-order Logic
- Knuth-Bendix style ordering guarantees termination of queries
- Beyond classification learning - strategy learning

Limitations

- Generalise beyond Dyadic logic
- Deal with classification noise
- Probabilistic Meta-Interpretive Learning
- Active learning