Inductive Programming
Lecture 4
Hypothesising an Algorithm from One Example

Stephen Muggleton
Department of Computing
Imperial College, London and
University of Nanjing

15th October, 2024

Papers for this lecture

Paper4.1: S.H. Muggleton. Hypothesising an algorithm from one example: the role of specificity. Philosophical Transaction of the Royal Society A, 381:20220046, 2023.

Motivation

- Inductive Programming
- Simple repetitive programs
- PAC, Blumer bound, Strong Learning Bias
- One-shot induction

Textual analogy problem

alice	ECILA
bert	?

Expected human response - human bias?

alice	ECILA
bert	TREB

One-Shot Learning - Open Question

Cognitive Science "People can learn .. concepts from just one example , but it remains a mystery how this is accomplished." (Lake et al, Proc Cognitive Science, 2011)

Relevant human background knowledge for learning Average human vocabulary - 10,000 - 42,000 (Goulden et al, 1990)

Key Question Under what circumstances can machines learn accurate hypotheses from one example?

Computer Science - Positive-only Learnability

Gold 1967: No infinite language, in the Chomsky hierarchy, can be exactly identified from a positive example sequence.

Valiant 1984: k-CNF propositional formulae can be learned efficiently (polynomial time) with high accuracy from a randomly selected positive example sequence.

Muggleton 1996: Given a Bayes' prior distribution over hypotheses, efficient (polynomial runtime) logic programs can be learned efficiently, with high expected accuracy, from a randomly selected positive example sequence.

Bayes' framework [Muggleton, 1996]

$D_{\mathcal{H}}$**:** probability distribution over hypothesis class $\mathcal{H}$.

D_X**:** probability distribution over instance class X.

$T \in \mathcal{H}$**:** teacher's target chosen randomly from $D_{\mathcal{H}}$.

$E = x_1 \ldots x_m$**:** examples of T chosen randomly from D_X.

$H \in \mathcal{H}$**:** learner's hypothesis.

$sz(H) = -ln D_{\mathcal{H}}(H)$**:** size of H.

$g(H) = \sum_{x \in H} D_X(x)$**:** generality of H.

Bayes' positive-only MAP selection

Muggleton, 1996

$$\begin{aligned} p(H|E) &= \frac{p(H)p(E|H)}{p(E)} \\ &= p(H)\left(\frac{1}{g(H)}\right)^m c_m \\ -ln\ p(H|E) &= sz(H) + m\,(ln\ g(H)) + d_m \end{aligned}$$

$$\text{Minimise } -ln\ p(H|E) \text{ over } H \in \mathcal{H}$$

One-shot Learning, m=1 case

$$-ln\ p(H|E) = sz(H) + \ ln\ g(H) + d_1$$

Key Finding

Source	Type	Expected Error
Muggleton, 1996	Pos only	$EE(m) \leq \frac{2.33+2ln\ m}{m}$
Muggleton, 1996	Pos+Neg	$EE(m) \leq \frac{1.51+2ln\ m}{m}$
One-shot	m=1 given g(T)	$EE(1\|g(T)) \leq 4.66g(T)$

$EA(1|g(T)) \geq 95\%$ when $g(T) \leq 0.01$

Expected accuracy below default

Accurate one-shot learning requires a low-generality target

DeepLog - two stage hypothesis construction

Meta-Compilation Examples used to find a minimal Input-Output transformation sequences. Each transformation is an application of a primitive relation from the library.

Meta-Interpretation For each example, a transformation sequence is threaded into the hypothesised logic program. The program size is constrained by a bound. The bound is varied to find a minimum program with low generality.

DeepLog - Regular Grammar

Target	Example ($\sigma \to \tau$)	Primitives P
abc4	$\langle$a,b,c,d,e,f,c,d,e,f,g,h$\rangle \to \langle\rangle$	Library 63 primitives

Output Hypothesis H [7]	Evaluation
abc4(X,Y) :- a(X,Z), abc4_1(Z,Y).	$Time = 0.15s/0.56s$
abc4_1(X,Y) :- b(X,Z), abc4_1_1(Z,Y).	
abc4_1_1(X,Y) :- g(X,Z), h(Z,Y).	$g(H) = \frac{1}{1080} \approx 0.0009$
abc4_1_1(X,Y) :- cdef(X,Z), abc4_1_1(Z,Y).	
Introduced Auxiliaries [3]	$-ln\ p(H\|E) = 17.97$
cdef(X,Y) :- cd(X,Z), ef(Z,Y).	
cd(X,Y) :- c(X,Z), d(Z,Y).	$EA(H) > 99.57\%$
ef(X,Y) :- e(X,Z), f(Z,Y).	

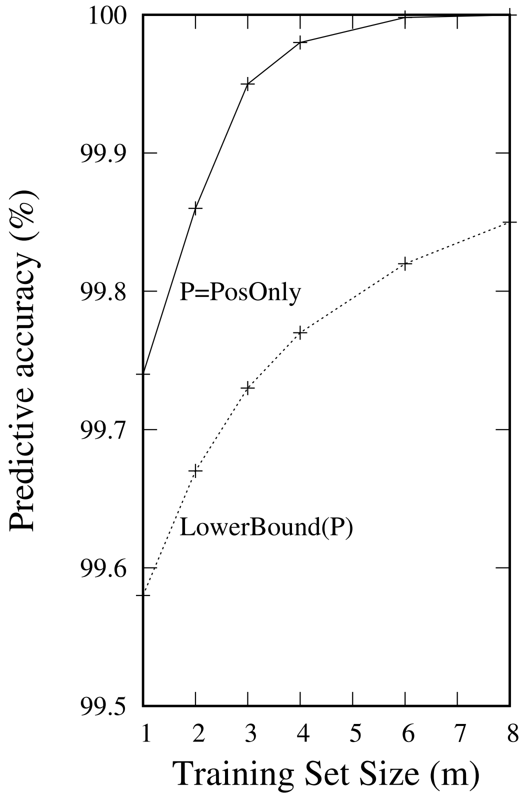
Calculating g(H)

abc4(X,Y) :- a(X,Z), abc4_1(Z,Y).	u=g(abc4)
abc4_1(X,Y) :- b(X,Z), abc4_1_1(Z,Y).	v=g(abc4_1)
abc4_1_1(X,Y) :- g(X,Z), h(Z,Y).	w=g(abc4_1_1)
abc4_1_1(X,Y) :- cdef(X,Z), abc4_1_1(Z,Y).	

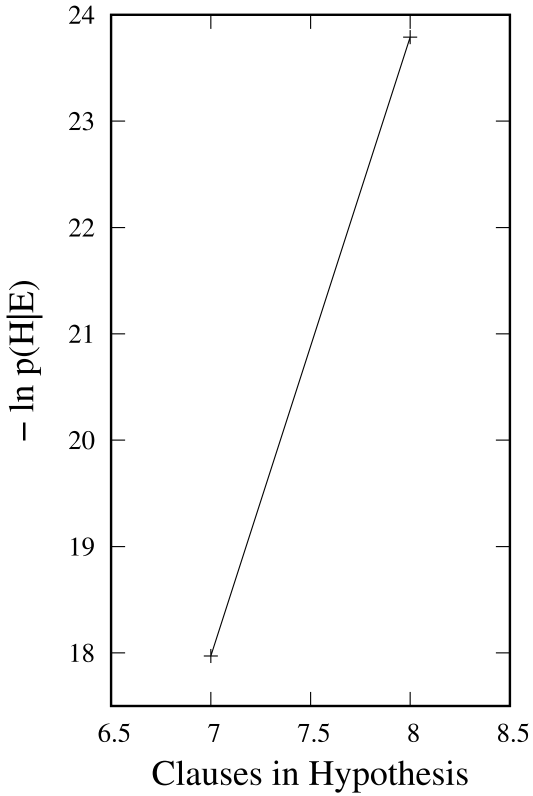
$u = \frac{v}{6}$, $v = \frac{w}{6}$, $w = \frac{1}{6}^2 + \frac{w}{6}$	Equations
$\Longrightarrow \; u = \frac{w}{36}$, $w = \frac{1}{36} + \frac{w}{6}$	for
$\Longrightarrow \; \frac{5w}{6} = \frac{1}{36}$	CLPR
$\Longrightarrow \; w = \frac{6}{180} = \frac{1}{30}$	solver
$\Longrightarrow \; u = \frac{1}{1080}$	

Regular Grammar Results

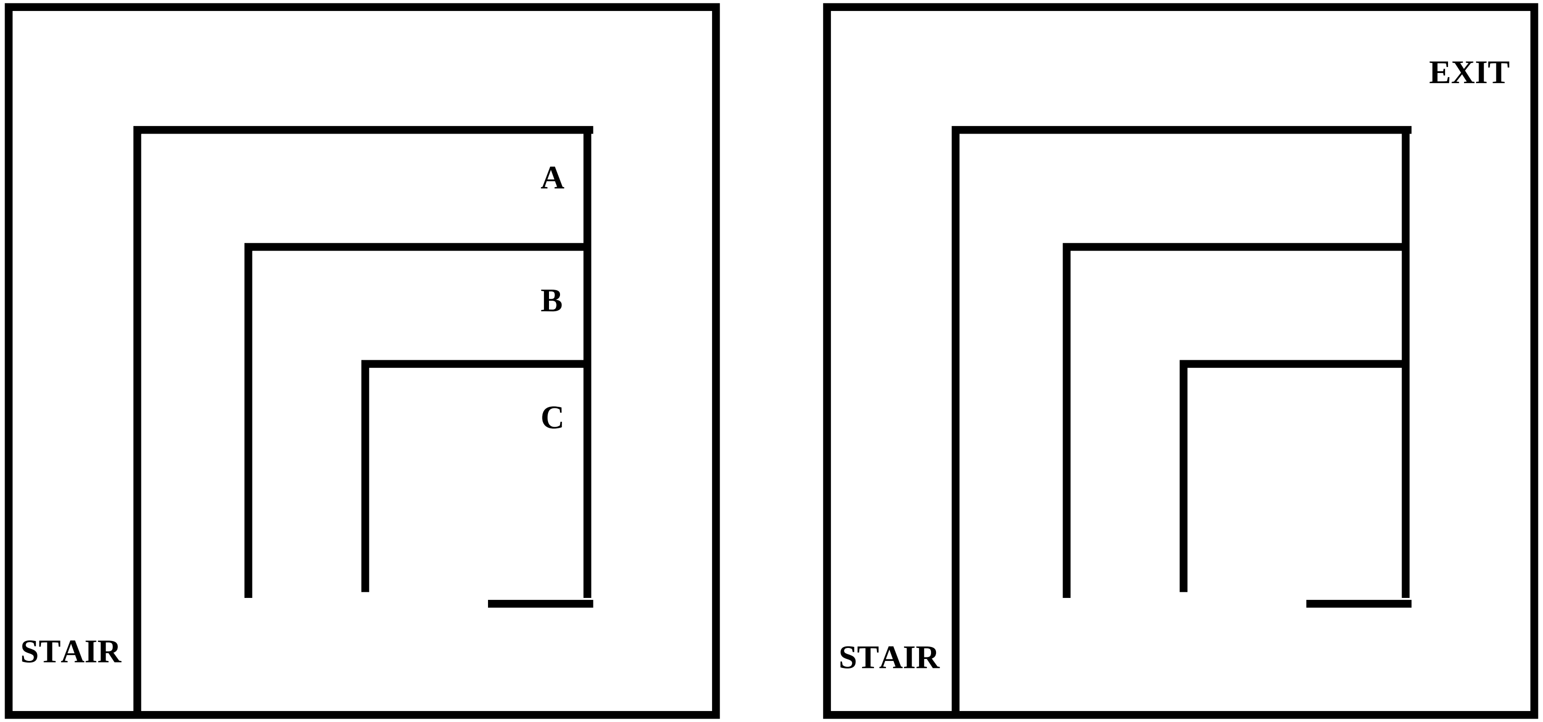
Accuracy



Posterior



Fire Escape 16 Storey Building Floorplans



FLOOR 16

FLOOR 1

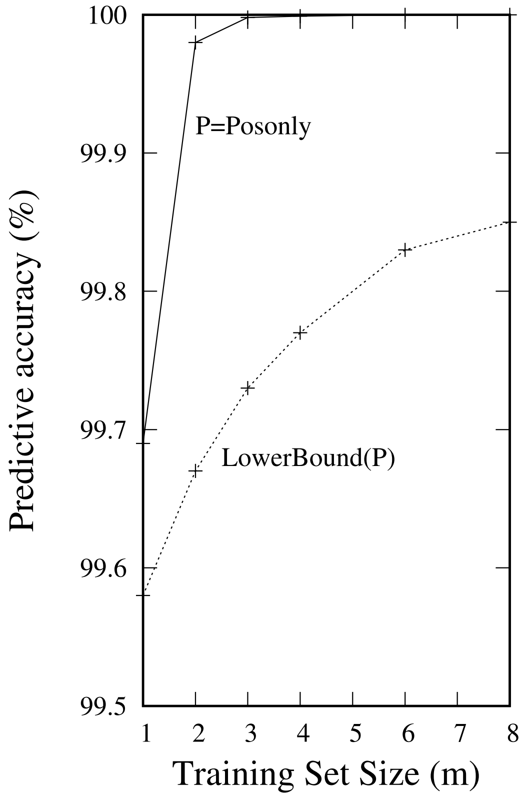
Fire Escape 16 Storey Building

Target	Example ($\sigma \rightarrow \tau$)	Primitives P
fire16	at(8,8,16) $\rightarrow$ at(10,10,1)	Library 63 primitives

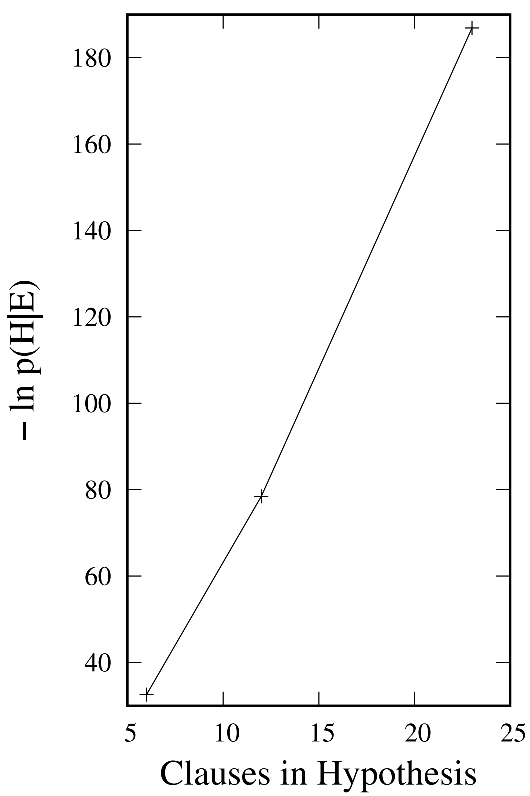
Output Hypothesis H [7]	Evaluation
fire16(X,Y) :- ws(X,Z), fire16_1(Z,Y). fire16_1(X,Y) :- ss(X,Z), fire16_1_1(Z,Y). fire16_1_1(X,Y) :- ns(X,Z), es(Z,Y). fire16_1_1(X,Y) :- d(X,Z), fire16_1_1(Z,Y). fire16_1_1(X,Y) :- es(X,Z), fire16_1_1_1(Z,Y). fire16_1_1_1(X,Y) :- ns(X,Z), fire16(Z,Y).	$Time = 0.2s/4.14s$ $g(H) = \frac{1}{1079} \approx 0.0009$ $-ln\ p(H\|E) = 32.57$ $EA(H) > 99.57\%$

Fire Escape 16 Results

Accuracy



Posterior

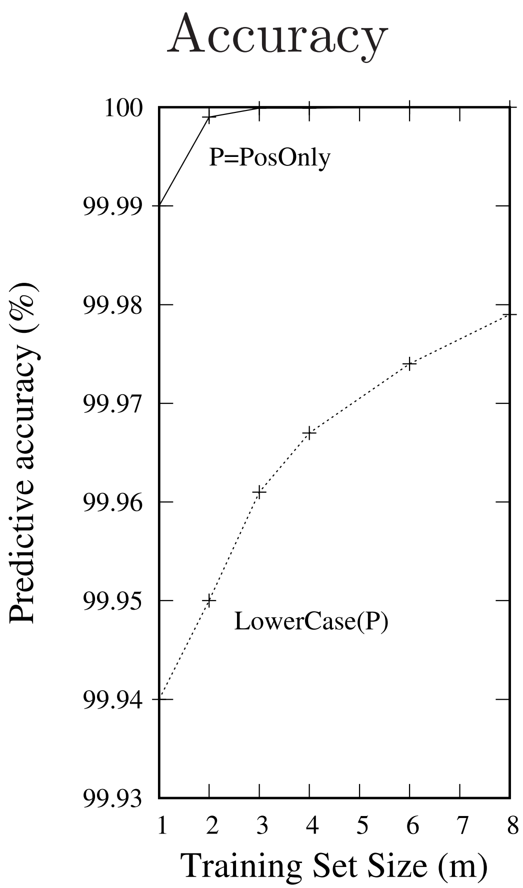


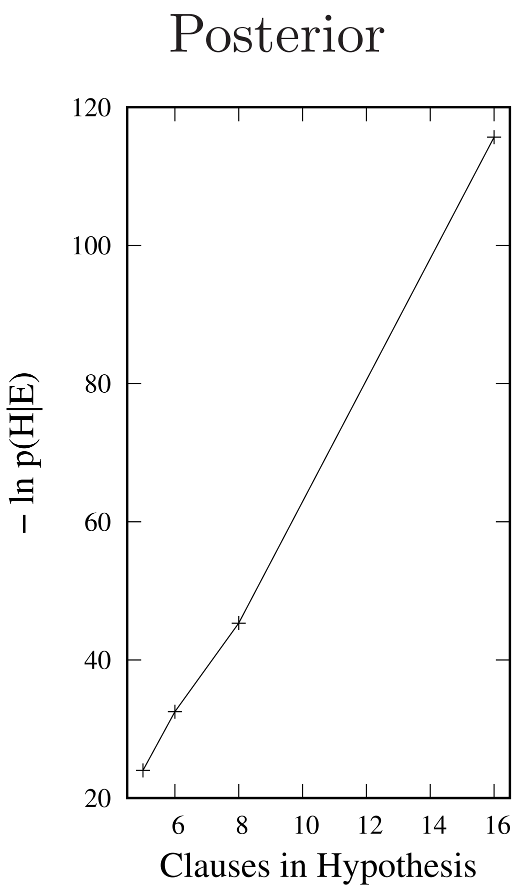
Reverse Uppercase

Target	Example ($\sigma \rightarrow \tau$)	Primitives P
rvup	$\langle$a,l,i,c,e$\rangle \rightarrow \langle$E,C,I,L,A$\rangle$	Library 63 primitives

Output Hypothesis H [5]	Evaluation
rvup(X,Y) :- call1(X,Z), rvup_1(Z,Y).	$Time = 0.21s/0.65s$
rvup_1(X,Y) :- pop(X,Z), rvup_1_1(Z,Y).	$g(H) = \frac{1}{7740} \approx 0.0001$
rvup_1_1(X,Y) :- upc(X,Z), rvup_1_1_1(Z,Y).	
rvup_1_1_1(X,Y) :- push(X,Z), return1(Z,Y).	$-ln\ p(H\|E) = 24.01$
rvup_1_1_1(X,Y) :- push(X,Z), rvup_1(Z,Y).	$EA(H) > 99.94\%$

Reverse Uppercase Results





Summary

One-shot learning Analogy problems show humans make single example hypotheses with high consensus. One-shot learning in Cognitive Science and Artificial Intelligence.

Bayes' model of One-shot learning Bayes model of One-shot learning special case of earlier positive-only model. Effectiveness for low-generality targets.

DeepLog Experiments with DeepLog show construction of high-accuracy general recursive programs from one example.

Further work Investigate circumstances in which an incrementally learned large background library supports rather than degrades further learning. What is the role of low-generality background primitives?