Learning MDL logic programs from noisy data

Céline Hocquette, Andreas Niskanen, Matti Järvisalo, Andrew Cropper

University of Oxford, University of Helsinki

UK Research and Innovation

Research Council of Finland

UNIVERSITY OF HELSINKI

UNIVERSITY OF OXFORD

Inductive Logic Programming

Inductive Logic Programming

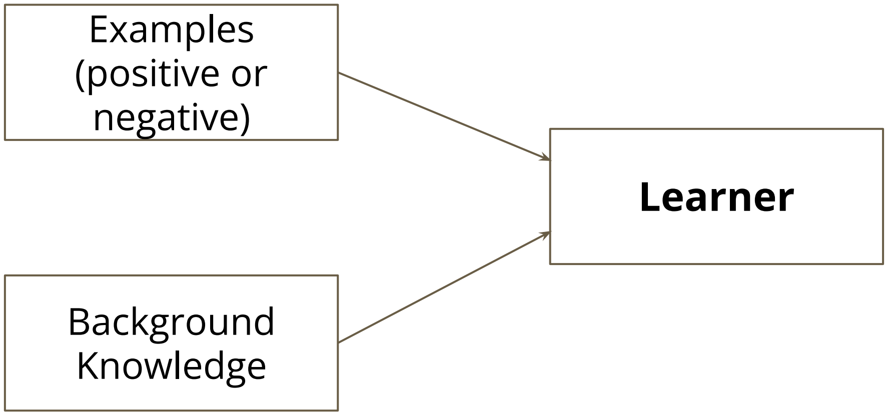
Examples (positive or negative)

Inductive Logic Programming

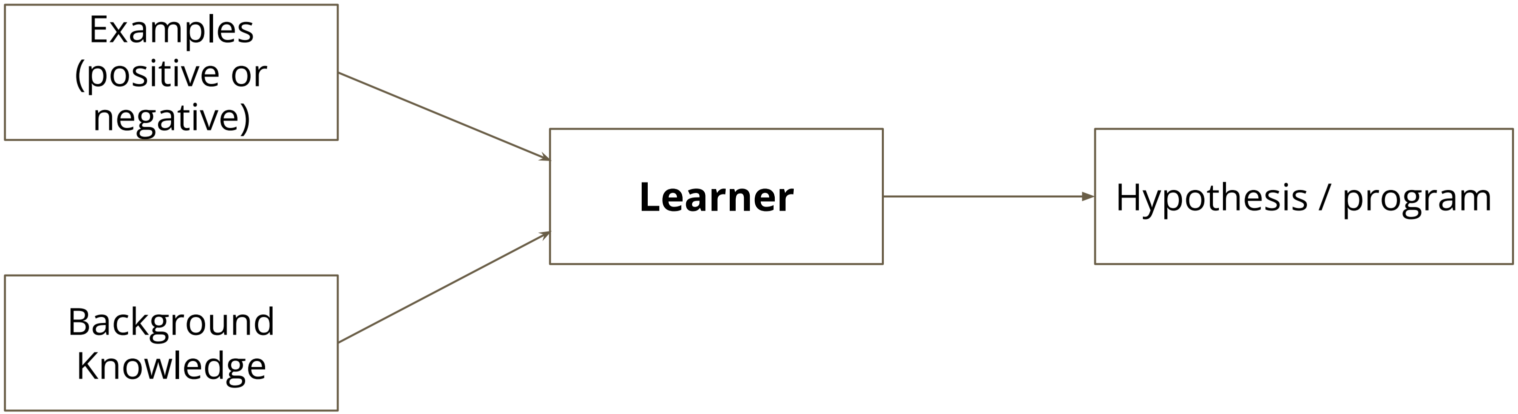
Examples (positive or negative)

Background Knowledge

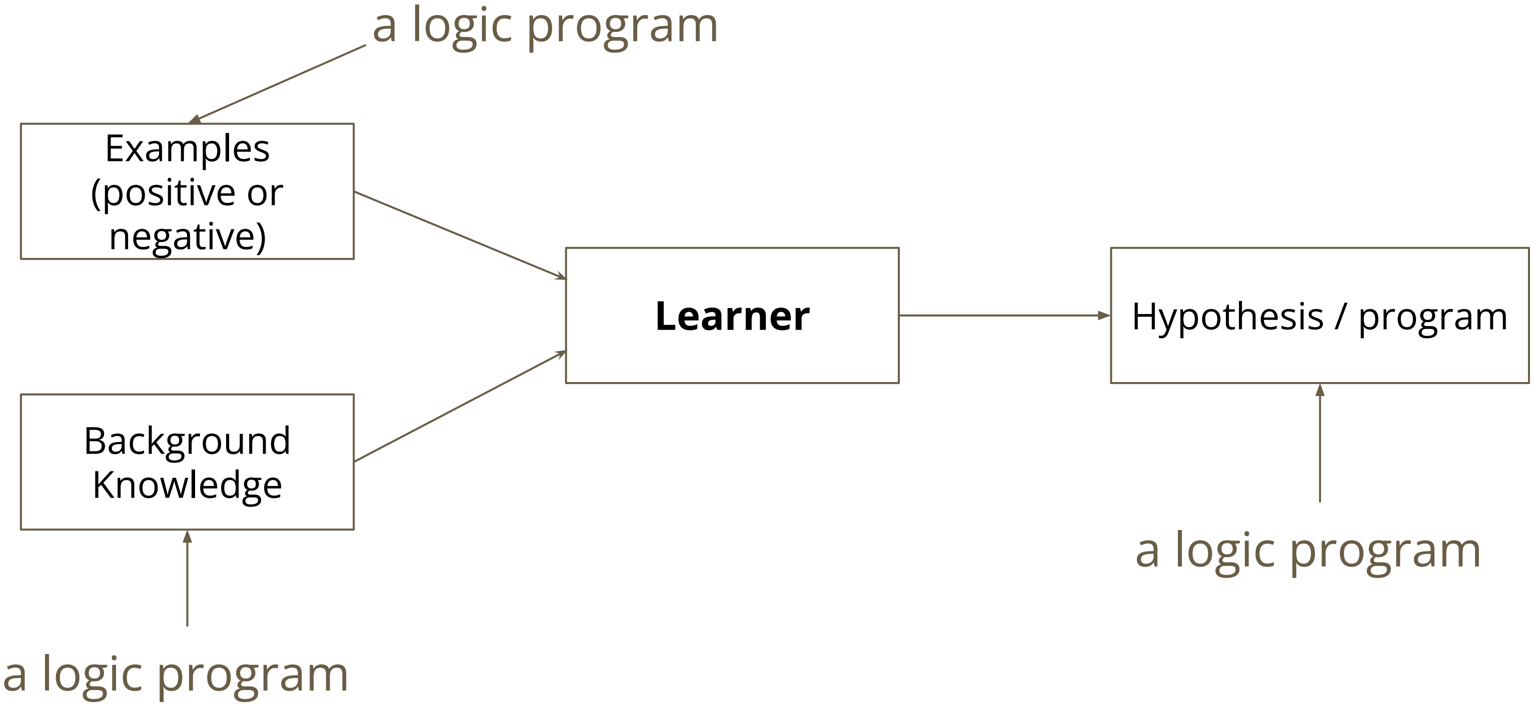
Inductive Logic Programming



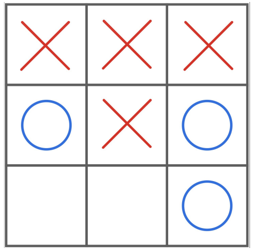
Inductive Logic Programming

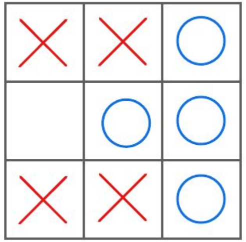


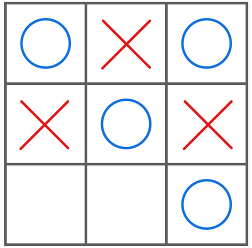
Inductive Logic Programming



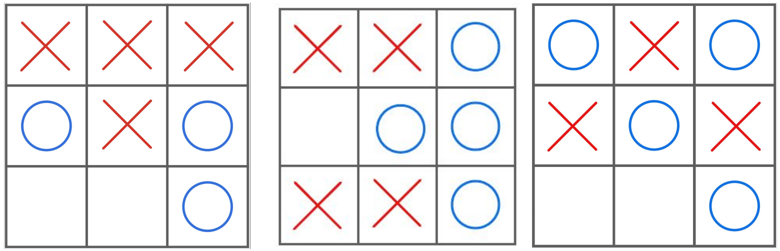
Positive



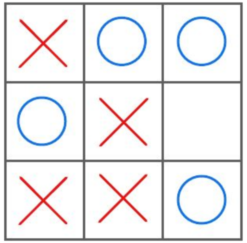




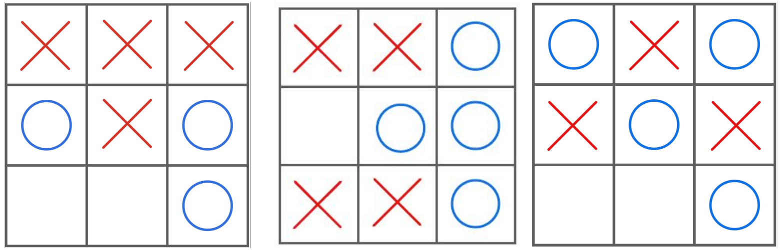
Positive



Negative

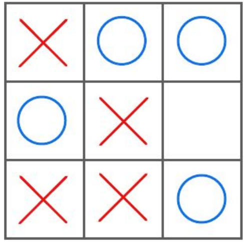


Positive



b1 b2 b3

Negative

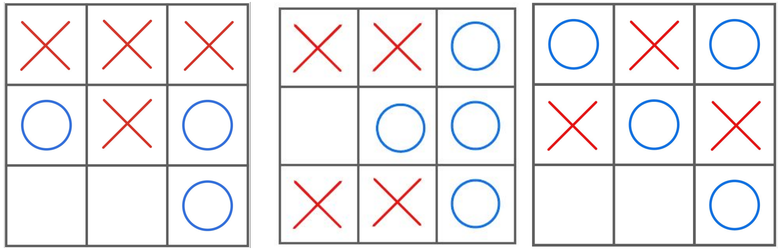


b4

Positive examples	Negative examples

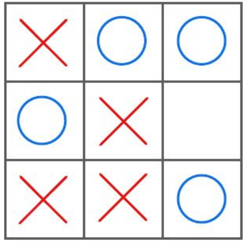
`win(b1,x).`
`win(b2,o).`
`win(b3,o).`
	`win(b4,x).`

Positive



b1 b2 b3

Negative



b4

Positive examples	Negative examples

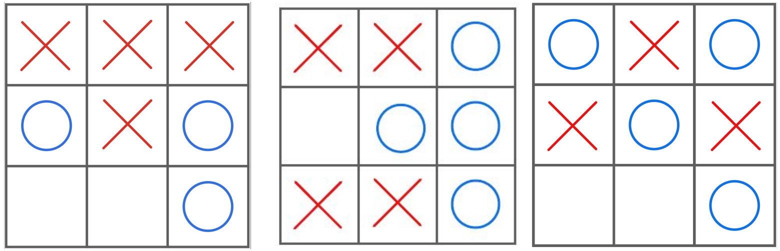
`win(b1,x).`
`win(b2,o).`
`win(b3,o).`
	`win(b4,x).`

Background Knowledge

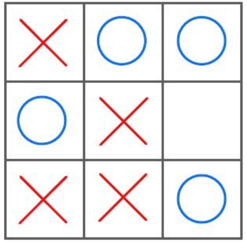
`cell(b1,0,x).`
`cell(b1,1,x).`
`cell(b1,2,x).`
…
`cell(b2,0,x).`
…
 	
`cell(b3,0,o).`

0	1	2
3	4	5
6	7	8

Positive



Negative



Program

```
win(Board,Player) ← cell(Board,0,Player), cell(Board,1,Player), cell(Board,2,Player)
win(Board,Player) ← cell(Board,2,Player), cell(Board,5,Player), cell(Board,8,Player)
win(Board,Player) ← cell(Board,0,Player), cell(Board,4,Player), cell(Board,8,Player)
```

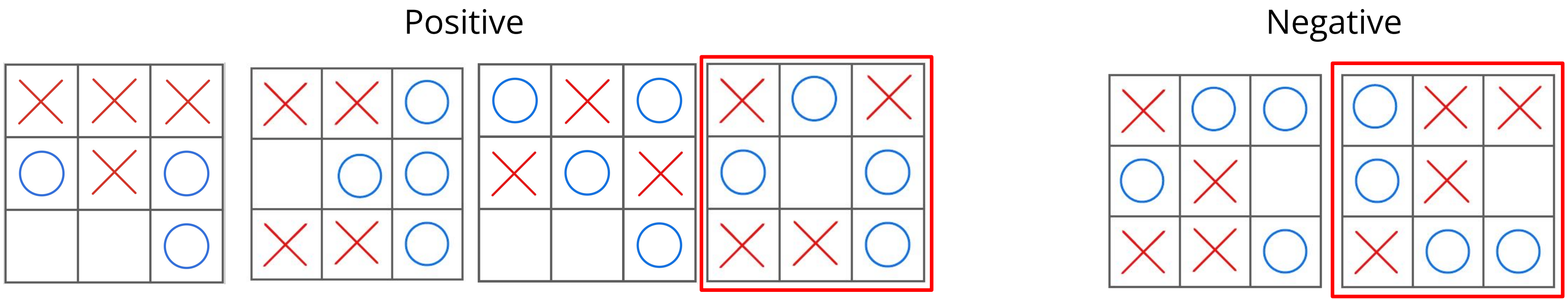
0	1	2
3	4	5
6	7	8

In this work

An approach to learn programs from noisy (mislabelled) examples

In this work

An approach to learn programs from noisy (mislabelled) examples



Challenges

Challenges

- unspecified level of noise ε

Challenges

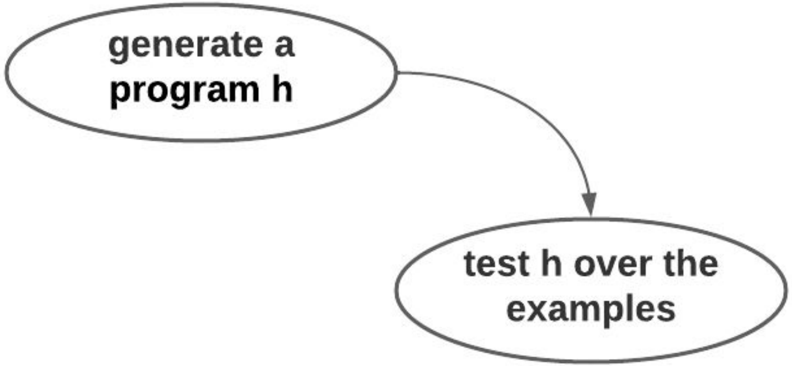
- unspecified level of noise ε
- overfitting

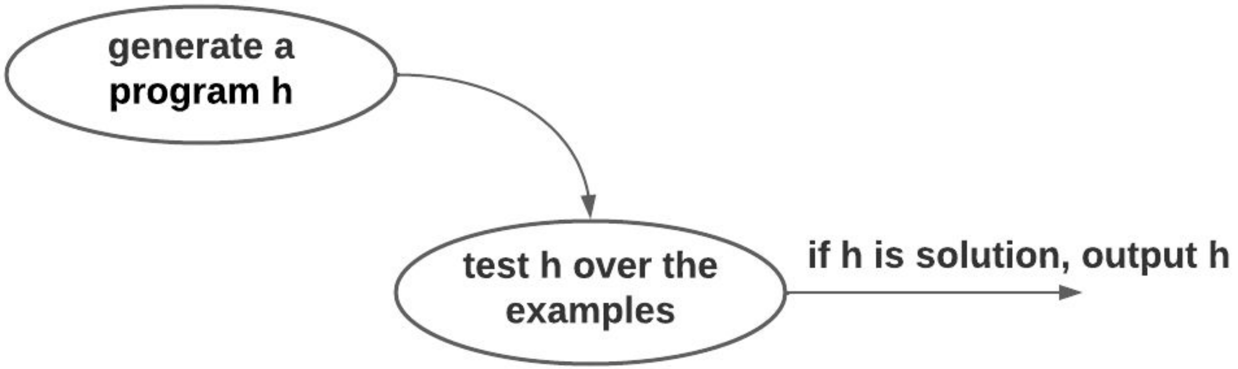
Challenges

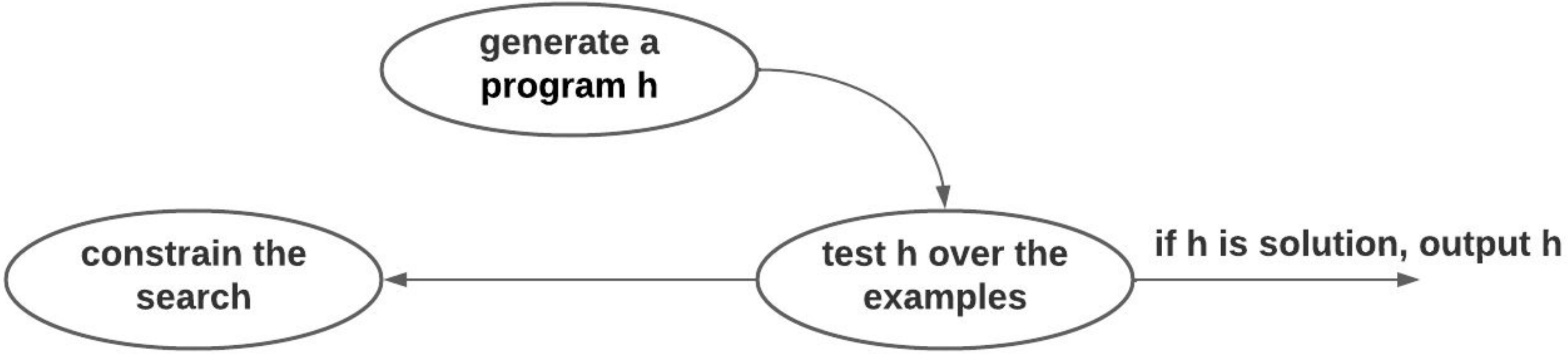
- unspecified level of noise ε
- overfitting
- learn complex programs (recursion and predicate invention)

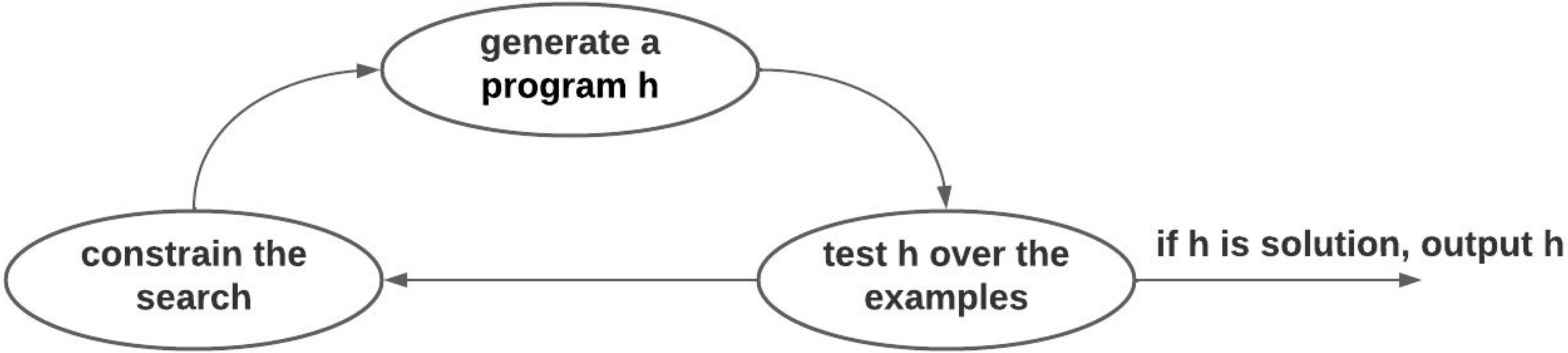
Our approach: based on learning from failures

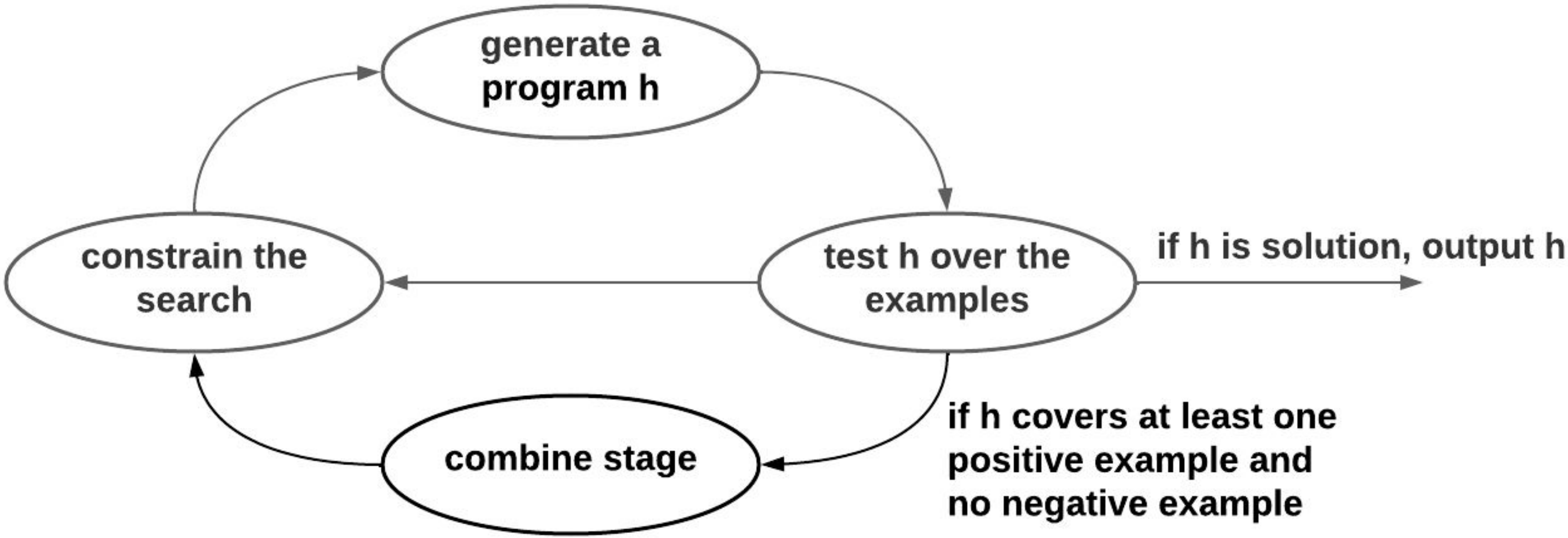
generate a
program h


generate a program h
test h over the examples


generate a program h
test h over the examples
if h is solution, output h


generate a program h
test h over the examples
if h is solution, output h
constrain the search


generate a program h
test h over the examples
if h is solution, output h
constrain the search


generate a
program h
constrain the
search
test h over the
examples
if h is solution, output h
if h covers at least one
positive example and
no negative example
combine stage

Existing approaches

Existing approaches

find a program which covers all positive examples, no negative examples and has minimal size

Our cost function

Our cost function

minimum description length: trade-off model complexity (program size) and data fit (training accuracy)

Our cost function

minimum description length: trade-off model complexity (program size) and data fit (training accuracy)

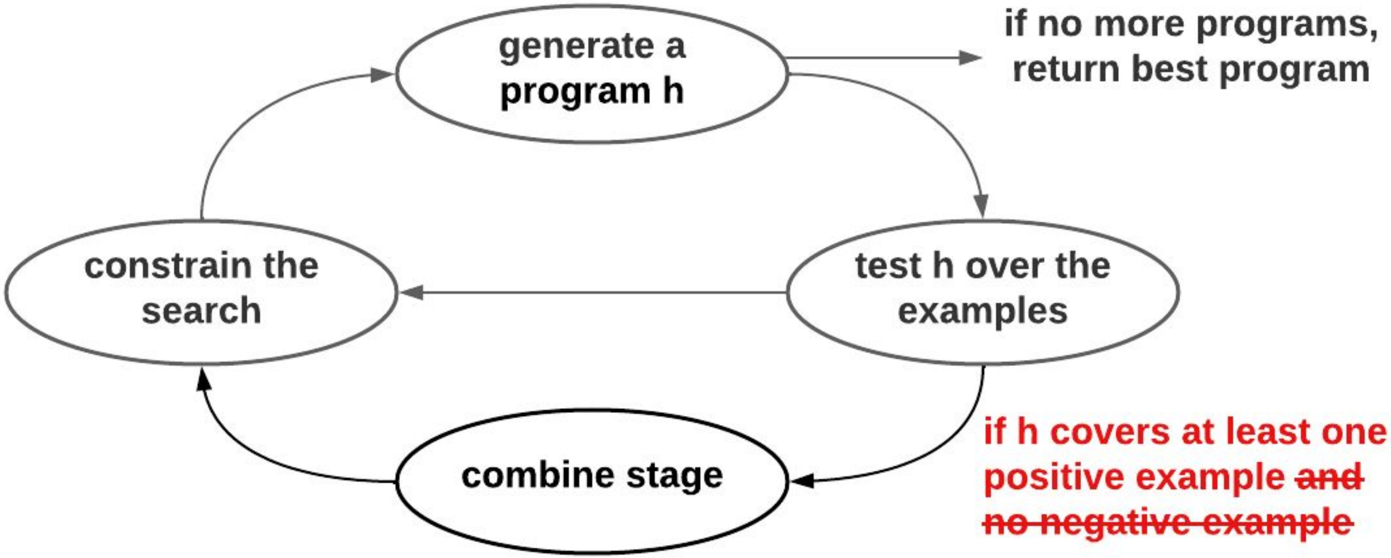
$$mdl(h) = size(h) + fp(h) + fn(h)$$

Our cost function

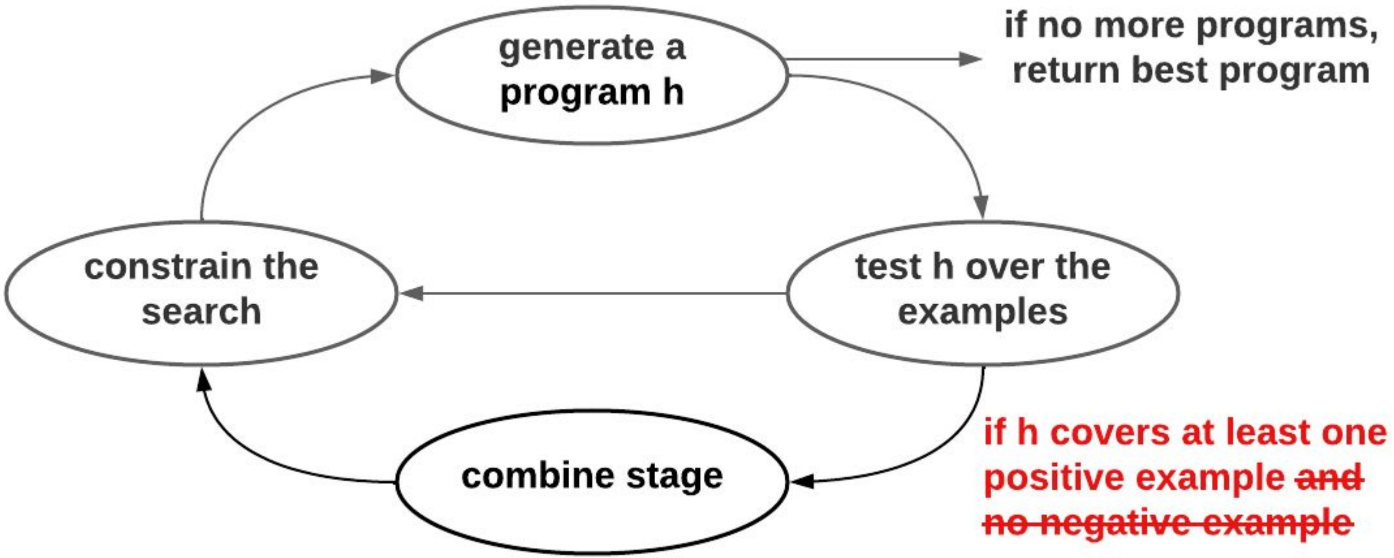
minimum description length: trade-off model complexity (program size) and data fit (training accuracy)

$$mdl(h) = Asize(h) + Bfp(h) + Cfn(h)$$
$$\text{with } A > 0, B \geq 0 \text{ and } C \geq 0$$

Our approach



Our approach



We use a MaxSAT solver to search for a MDL combination (a union) of programs.

Implementation

We implement our approach in MaxSynth.

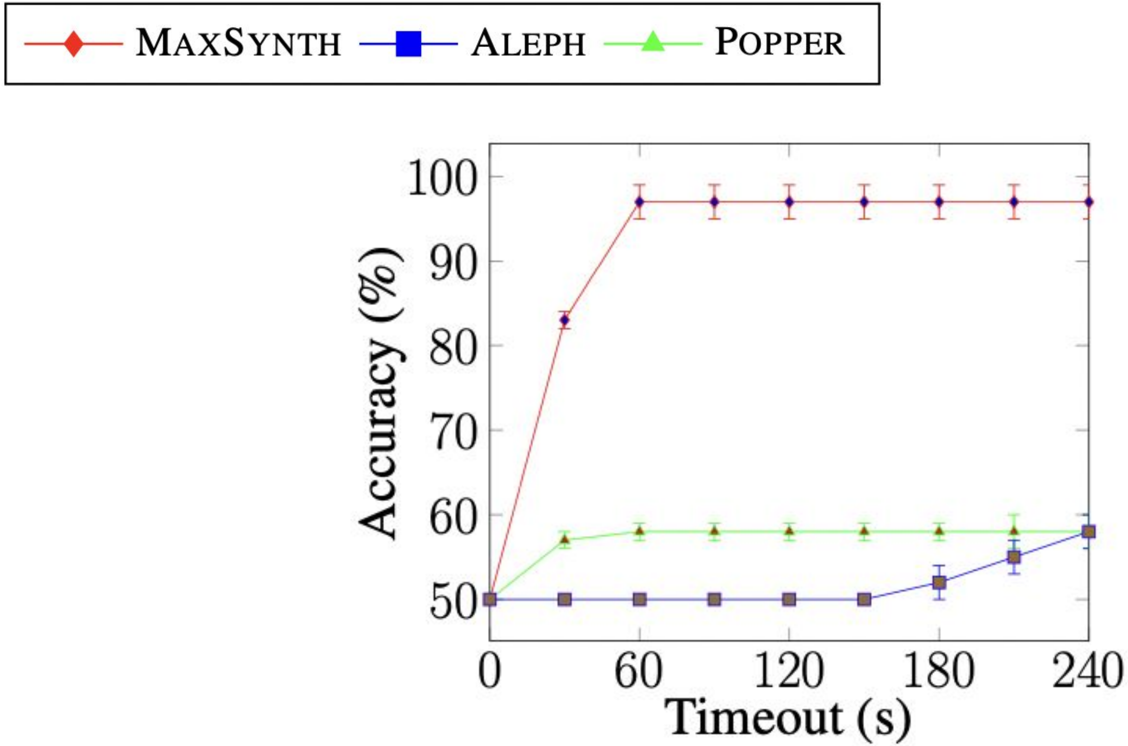
Implementation

We implement our approach in MaxSynth.

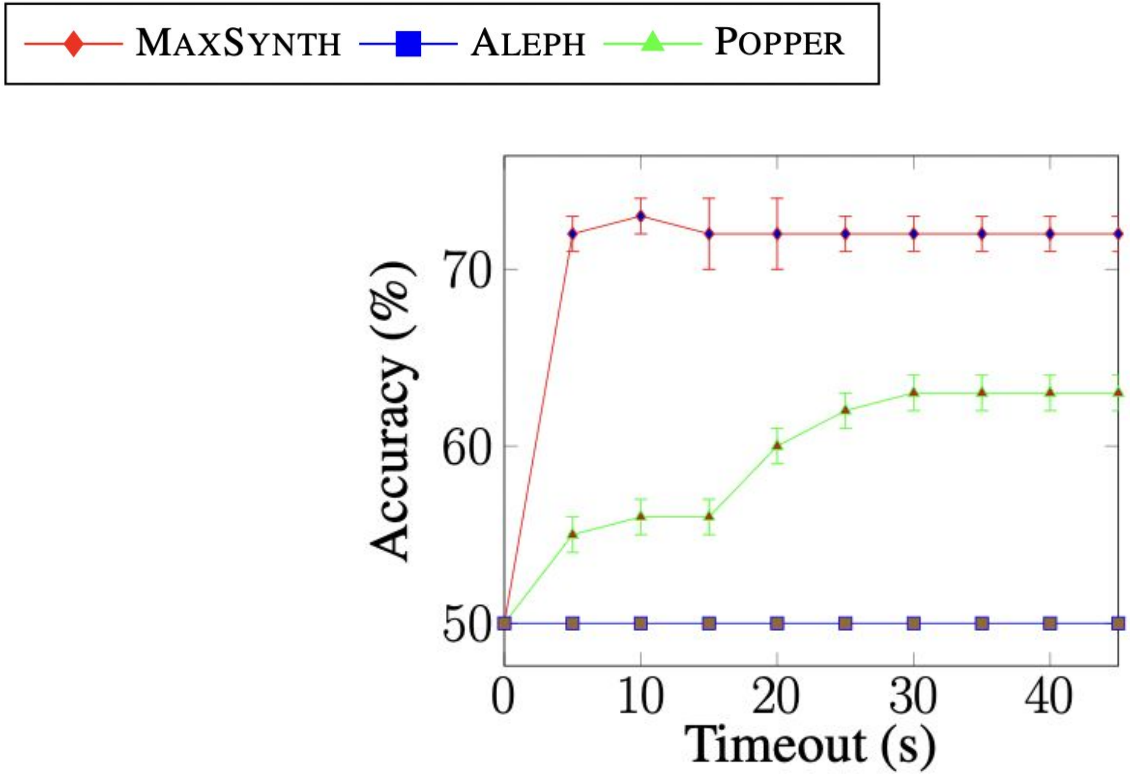
Theorem: MaxSynth learns an optimal solution (a MDL program) if one exists.

Does it work?

Q1 Can MAXSYNTH learn programs from noisy data?


MaxSynth
Aleph
Popper
100
90
80
70
60
50
Accuracy (%)
0
60
120
180
240
Timeout (s)

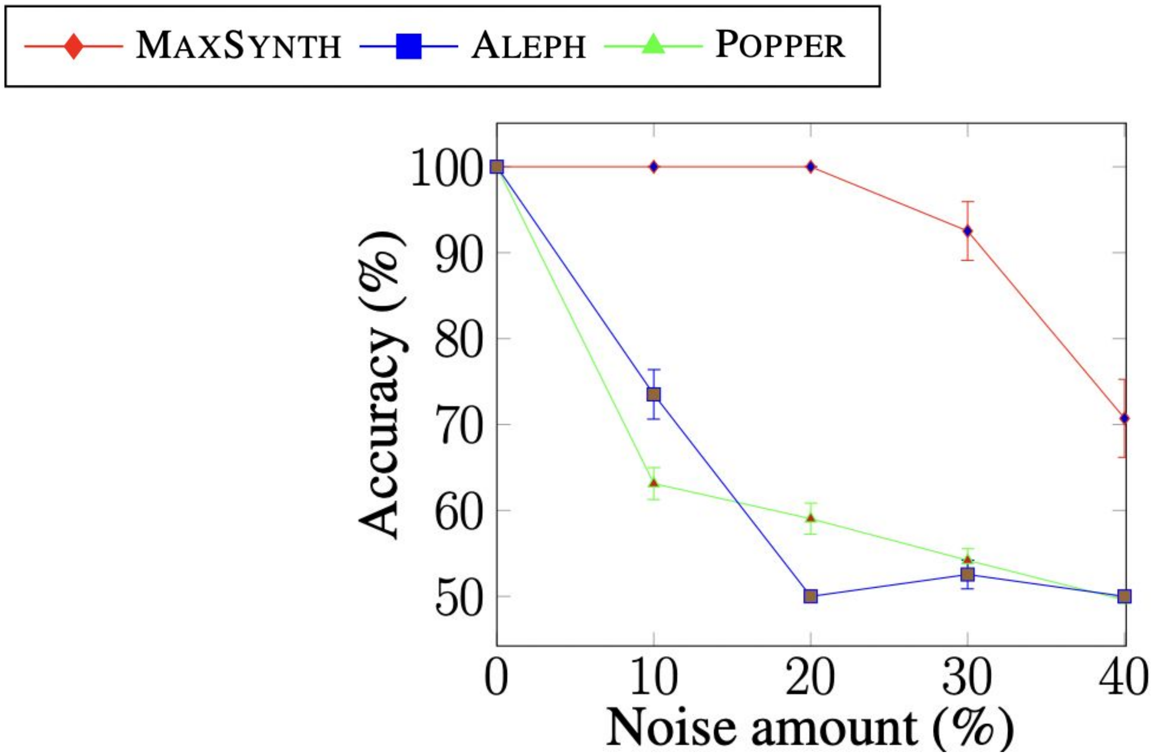
zendo



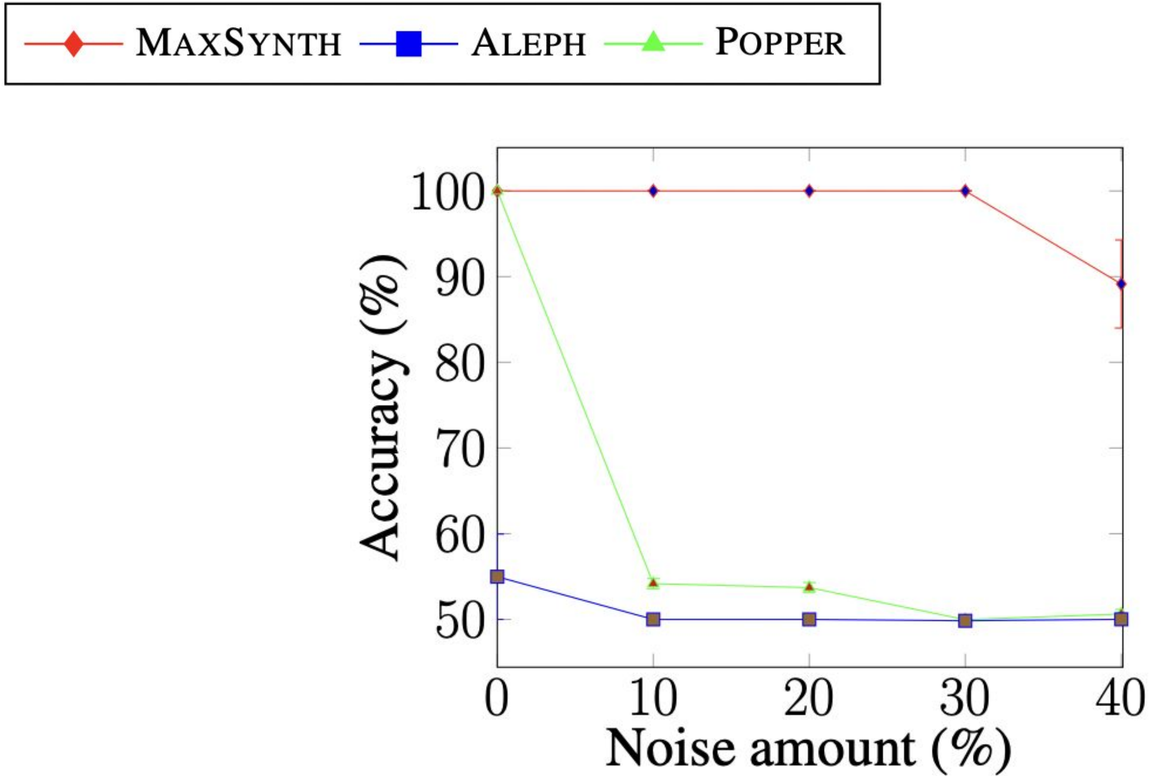
alzheimer-toxic

Does it work?

Q2 How well does MAXSYNTH handle progressively more noise?


MaxSynth
Aleph
Popper
Accuracy (%)
100
90
80
70
60
50
0
10
20
30
40
Noise amount (%)

iggp-rps


MaxSynth
Aleph
Popper
Accuracy (%)
100
90
80
70
60
50
0
10
20
30
40
Noise amount (%)

dropk

Conclusion

An approach that learns **minimal description length** programs from **noisy** examples.

Our approach can:
- improve learning performance,
- scale to moderate amount of noise.

Limitation

- Cost function

$$mdl(h) = Asize(h) + Bfp(h) + Cfn(h)$$

$$\text{with } A > 0, B \geq 0 \text{ and } C \geq 0$$

Thank you!

Questions?