Meta-interpretive learning of data transformation programs

Andrew Cropper, Alireza Tamaddoni-Nezhad, Stephen H. Muggleton
Imperial College London

Input

P_011
67 year
lung disease: n/a, Diagnosis: Unknown
80.78%

P_003
56
Diagnosis: carcinoma, lung disease: unknown
20.78

P_013
70
Diagnosis: pneumonia
55.9

Output

P_011	67	Unknown
P_003	56	carcinoma
P_013	56	pneumonia

- Semi-structured
- Positive only learning
- Background knowledge

Input

P_011
67 year
lung disease: n/a, Diagnosis: Unknown
80.78%

P_003
56
Diagnosis: carcinoma, lung disease: unknown
20.78

P_013
70
Diagnosis: pneumonia
55.9

Output

P_011	67	Unknown
P_003	56	carcinoma
P_013	56	pneumonia

```
f(A,B):- f2(A,C), f1(C,B).
f2(A,B):- find_patient_id(A,C), find_int(C,B).
f1(A,B):- open_interval(A,B,[':',' ‘],[",'n']).
f1(A,B):- open_interval(A,B,[':',' '],[',',' ']).
```

MetagolD

Implementation of meta-interpretive learning*, a form of inductive logic programming based on a Prolog meta-interpreter, which supports predicate invention and the learning of recursive theories

* S.H. Muggleton, D. Lin, and A. Tamaddoni-Nezhad. Meta-interpretive learning of higher-order dyadic datalog: Predicate invention revisited. Machine Learning, 100(1):49-73, 2015.

Transformation language

- find_sublist/3
- closed_interval/4
- open_interval/4

open_interval/4 and closed_interval/4

```
Input = [i,n,d,u,c,t,i,o,n],
Start = [n,d],
End = [t,i]

open_interval(Input,[u,c],Start,End).
closed_interval(Input,[n,d,u,c,t,i],Start,End).
```

Experiment: ecological papers

Input

Harpalus rufipes eats large prey such as Lepidoptera

Bembidion lampros. In cereals the main food was Collembola

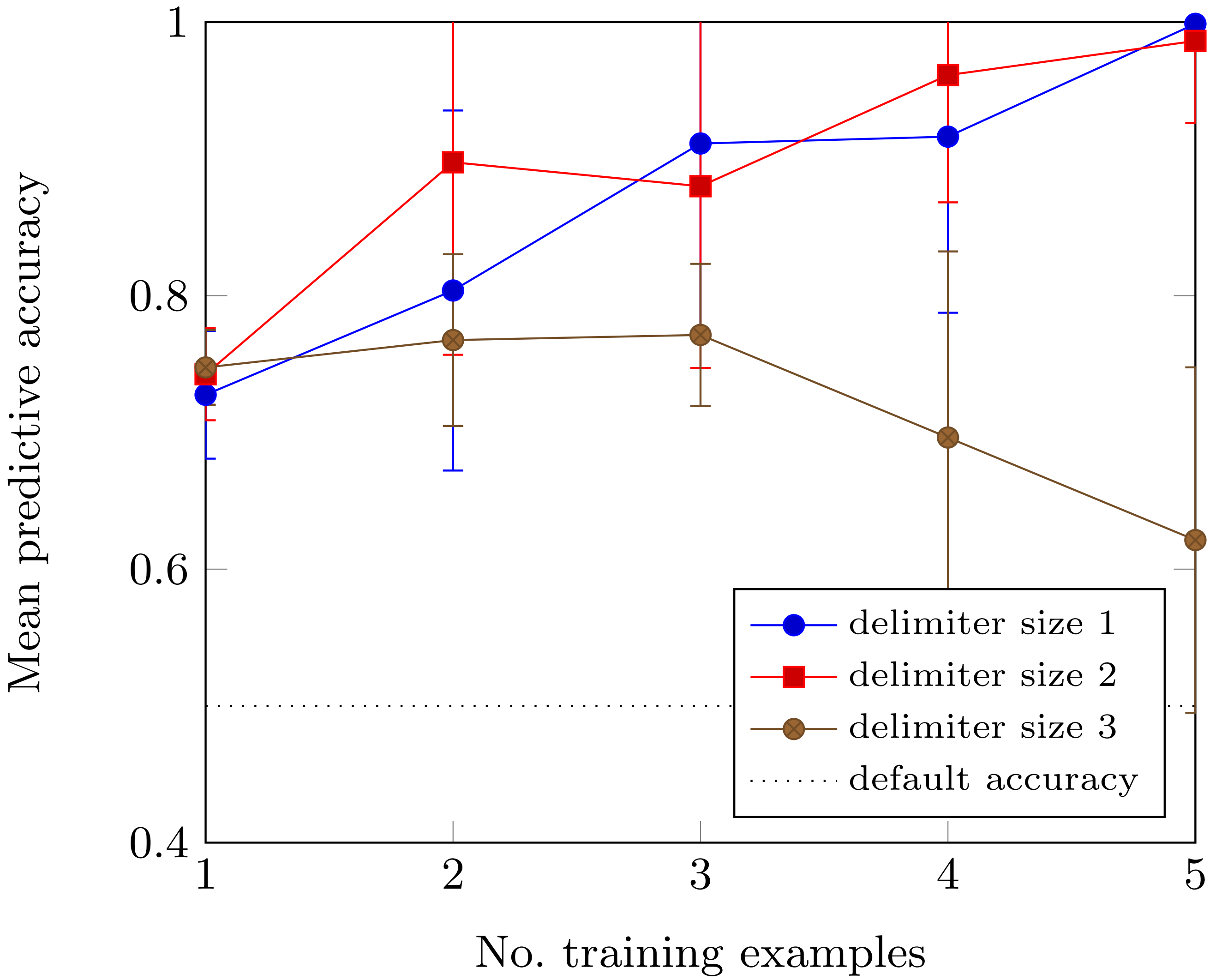
Output

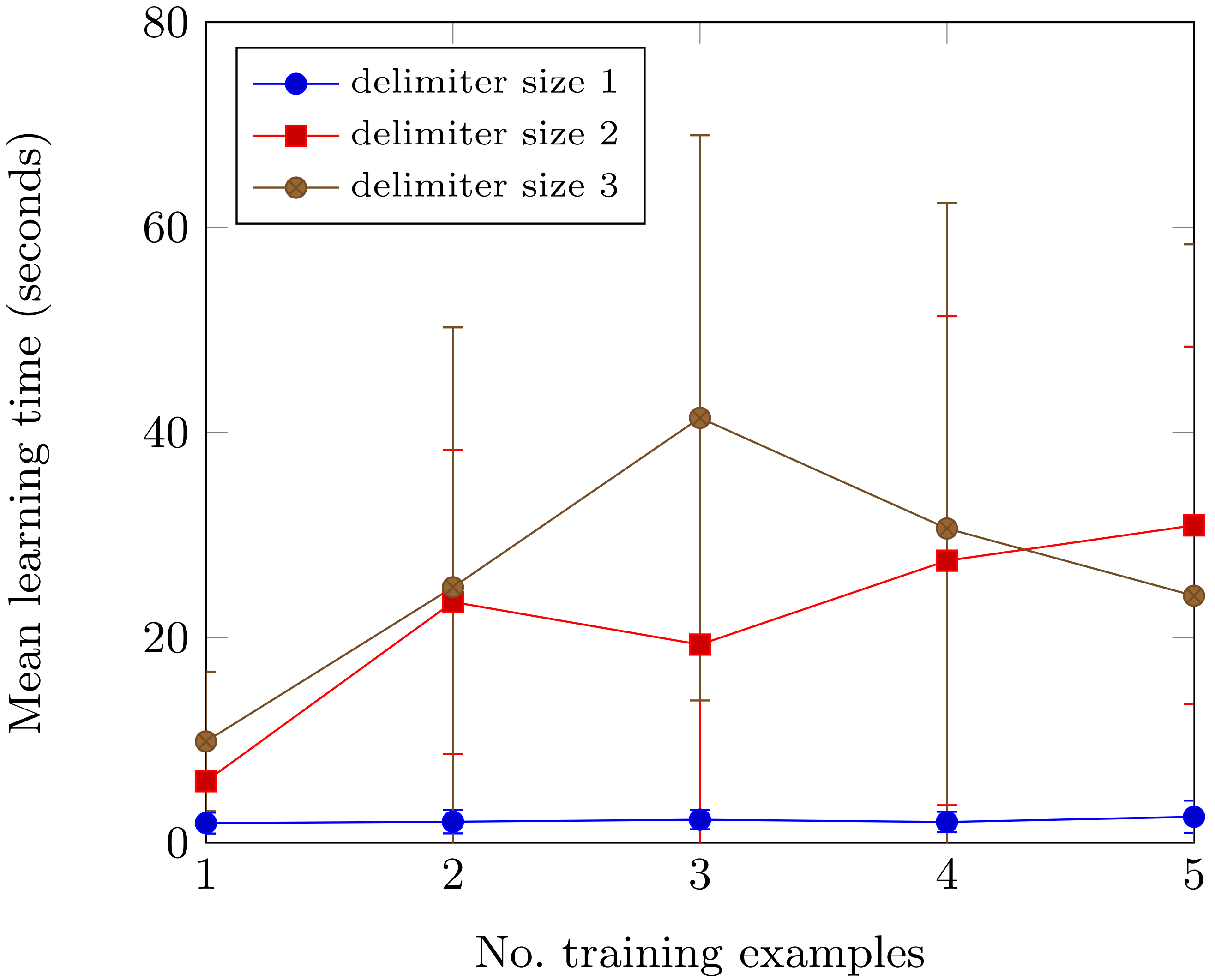
Harpalus rufipes	eats	Lepidoptera
Bembidion	food	Collembola

Learned program

```
f(A,B):- f3(A,C), find_species(C,B).
f3(A,B):- find_species(A,C), f2(C,B).
f2(A,B):- closed_interval(A,B,[f,o],[o,d]).
f3(A,B):- find_species(A,C), f1(C,B).
f1(A,B):- closed_interval(A,B,[e,a],[t,s]).
```

Experiment: ecological papers





Experiment: medical records

Input

P_011
67 year
lung disease: n/a, Diagnosis: Unknown
80.78%

P_003
56
Diagnosis: carcinoma, lung disease: unknown
20.78

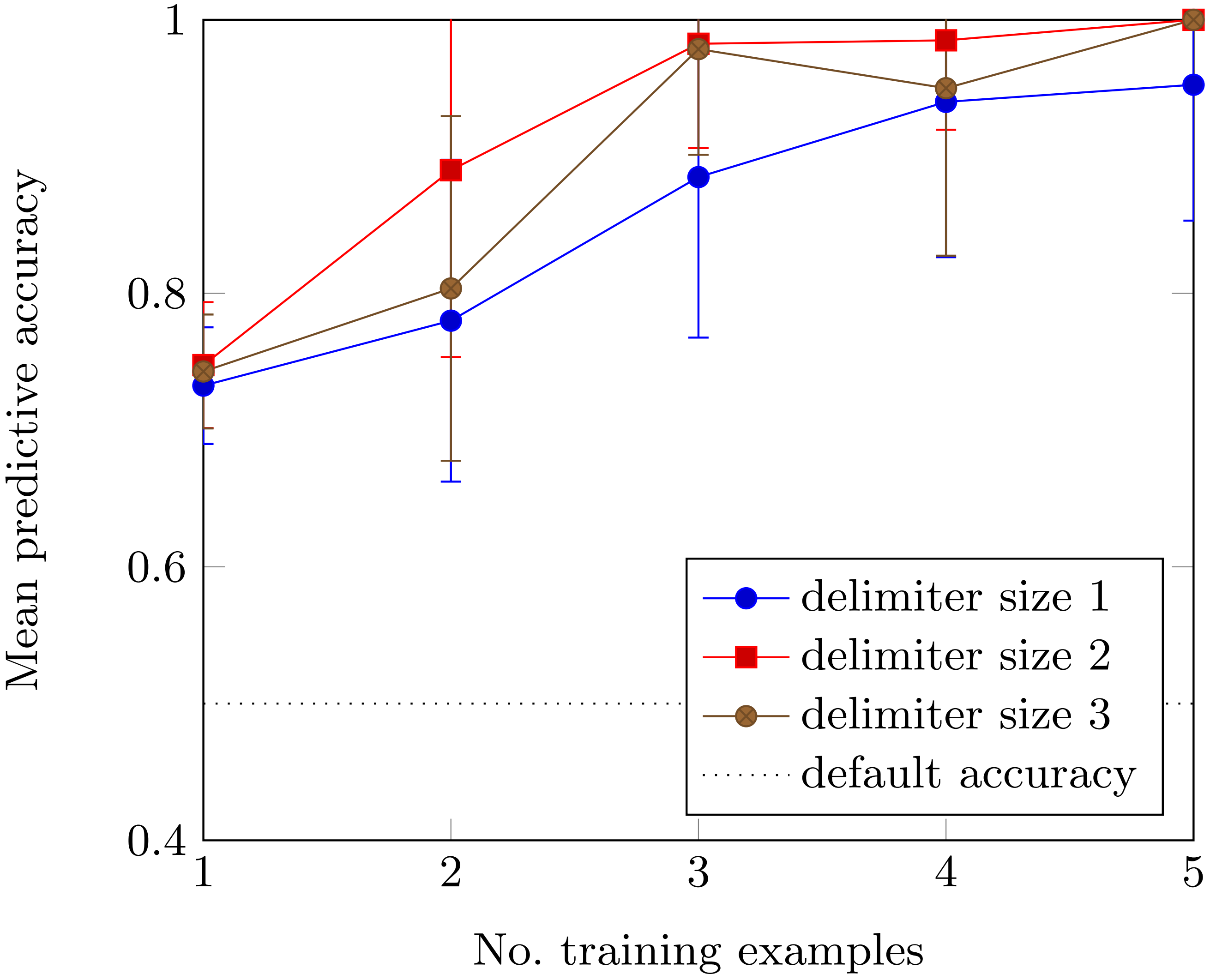
P_013
70
Diagnosis: pneumonia
55.9

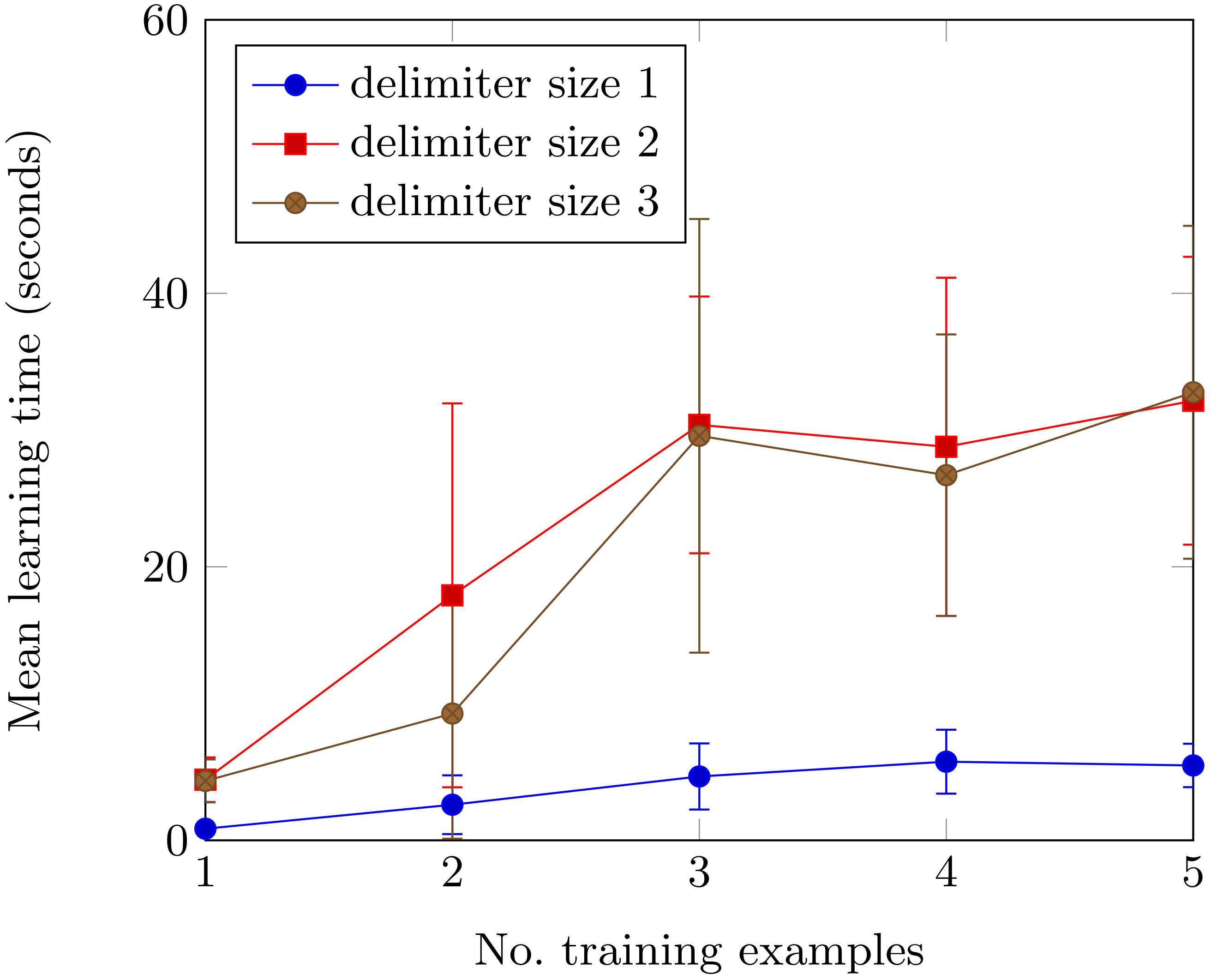
Output

P_011	67	Unknown
P_003	56	carcinoma
P_013	56	pneumonia

```
f(A,B):- f2(A,C), f1(C,B).
f2(A,B):- find_patient_id(A,C), find_int(C,B).
f1(A,B):- open_interval(A,B,[':',' ‘],['','n']).
f1(A,B):- open_interval(A,B,[':',' '],[',',' ']).
```

Experiment: medical records





Conclusions

- MIL is able to generate accurate data transformation programs from a small number of examples
- Delimiter size effects learning performance

Future work

- Apply to problems which require recursion
- Generate hypotheses in a scripting language
- Probabilistic approaches / noise handling

Thank you