Learning Brave Assumption-Based Argumentation Frameworks via ASP

Emanuele De Angelis, Cnr-IASI, Italy

Maurizio Proietti, Cnr-IASI, Italy

Francesca Toni, IMPERIAL, UK

Imperial College London
Department of Computing
European Research Council
Established by the European Commission
The Royal Society
Royal Academy of Engineering
J.P.Morgan

1. Unifying formalism
for various forms of
non-monotonic reasoning

LEARNING

2. Defeasible knowledge,
subject to argumentative debate

ASSUMPTION-Based ARGUMENTATION Frameworks

via ASP

4. Argumentative XAI
AI models explained using
argumentative explanations

3. Structured arguments
deductions from
assumptions using rules

LEARNING BRAVE

UMPTION-Ba

ATION F

via ASP

Automating learning
of ABA frameworks from
background knowledge
+
positive & negative examples

Goal

Implemented using
Answer Set Programming

Novel learning algorithm for
brave reasoning
under **stable extensions**
based on **transformation rules**

- Rote Learning
- Folding
- Assumption Introduction
- Subsumption

Contribution

ABA FRAMEWORKS

Assumptions

Language

$$\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$$

Rules

$$\underbrace{s}_{\textbf{head}} \leftarrow \underbrace{s_1, \ldots, s_n}_{\textbf{body}}$$

if $n=0$, $h \leftarrow$ is called **fact**

total mapping from $\mathcal{A}$ into $\mathcal{L}$

$\bar{a}$ is the **contrary** of a in $\mathcal{A}$

ABA FRAMEWORKS - SEMANTICS

a variant of the Nixon diamond problem

facts

```
quaker(a) ←     republican(a) ←
quaker(b) ←     republican(b) ←
```

assumptions

contrary of normal_quaker

contrary of normal_republican

```
pacifist(X) ← quaker(X), normal_quaker(X)
abnormal_quaker(X) ← republican(X), normal_republican(X)
abnormal_republican(X) ← quaker(X), normal_quaker(X)
```

rules

- **Arguments** are **deductions** of sentences using **rules** and supported by **assumptions**

  { normal_quaker(a) } ⊢ pacifist(a)

- **Attacks** are directed at the assumptions in the support of arguments

  { normal_republican(a) } ⊢ abnormal_quaker(a)

  attacks

  { normal_quaker(a) } ⊢ abnormal_republican(a)

- We focus on **stable extensions**

  any set of arguments S that
  1. do not attack each other (conflict-free)
  2. S attacks all arguments it does not contain

BRAVE LEARNING PROBLEM

Given

1. ABA framework $\mathcal{F} = < \mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\ } >$ (**background knowledge**) with at least one stable extension
2. **Ep** = { **positive examples** }
3. **En** = { **negative examples** }
4. **T** = { **learnable predicates** }

find $\mathcal{F}' = < \mathcal{L}', \mathcal{R}', \mathcal{A}', \bar{\ }' >$ with a stable extension S such that

i. $\mathcal{F} \subseteq \mathcal{F}'$
ii. every positive has an argument in S
iii. no negative has an argument in S

$\mathcal{F}'$ is a **solution** of the ABA learning problem

BRAVE LEARNING VIA TRANSFORMATION RULES

Learning ABA frameworks relies upon a set of **transformation rules**

$$\langle \mathcal{L}_1, \mathcal{R}_1, \mathcal{A}_1, \overline{}^{1} \rangle \longrightarrow \langle \mathcal{L}_2, \mathcal{R}_2, \mathcal{A}_2, \overline{}^{2} \rangle \longrightarrow \dots \longrightarrow \langle \mathcal{L}_n, \mathcal{R}_n, \mathcal{A}_n, \overline{}^{n} \rangle$$

background knowledge **intensional solution**

learnt rules do not make explicit reference to specific values in the universe

$\longrightarrow \in \{$ **Rote Learning**, **Folding**, **Assumption Introduction**, **Subsumption** $\}$

A **strategy** controls the order of application of the transformation rules

ASP-ABAlearnB

ASP-ABALearnB at work

on a variant of the Nixon diamond problem

Language
{quaker(X)←, republican(X)←, normal_quaker(X)}

Assumptions
{normal_quaker(X)}

$X \in \{a,b\}$

1. Background Knowledge $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$

Rules
{ quaker(a)←, quaker(b)←,
republican(a)←, republican(b)←,
pacifist(X)← quaker(X), normal_quaker(X) }

Contraries
{ $\overline{\text{normal\_quaker(X)}}$ = abnormal_quaker(X) }

2. Positive examples **Ep** = { pacifist(a) }
3. Negative examples **En** = { pacifist(b) }
4. Learnable Predicates **T** = { pacifist, abnormal_quaker }

LEARNING RULES AT WORK
ROTE LEARNING

Add **facts**

- from positive examples
- for contraries of assumptions

to get a (**non-intensional**) **solution**

It’s enough to learn

```
abnormal_quaker(X) ← X=b
```

to get

$$\mathcal{R}' = \mathcal{R} \cup \{\ \texttt{abnormal\_quaker(X)} \leftarrow \texttt{X=b}\ \}$$

⊎

```
pacifist(X)← quaker(X), normal_quaker(X)
```

LEARNING RULES AT WORK
FOLDING

Towards an **intensional** solution …

Generalise

```
abnormal_quaker(X) ← X=b
```

to

```
abnormal_quaker(X) ← republican(X)
```

by using

```
republican(X) ← X=b
```

WARNING
It also constructs an argument that **attacks** a positive example

RULES AT WORK
ASSUMPTION INTRODUCTION

Repairing the ABA framework to get a solution …

Add an **assumption** to avoid

- **attacking** a positive example
- **accepting** a negative example

```
abnormal_quaker(X) ← republican(X), normal_republican(X)
```

NEW ASSUMPTION
with contrary
`abnormal_republican(X)`

AND REPEAT!

Rote Learning	`abnormal_republican(X) ← X=a`
Folding	`abnormal_republican(X) ← quaker(X)`
Assumption introduction	`abnormal_republican(X) ← quaker(X), normal_quaker(X)`

reuse

`pacifist(X) ← quaker(X), normal_quaker(X)`

`normal_quaker(X)`
is "relative to"
`quaker(X)`

No more contraries to learn:
LEARNING COMPLETED!

A GLIMPSE OF IMPLEMENTATION ROTE LEARNING via ASP

- **ASP encoding**

```
pacifist(X) :- quaker(X), normal_quaker(X).
normal_quaker(X) :- quaker(X), not abnormal_quaker(X).
{ abnormal_quaker(X) } :- quaker(X).
#minimize{1,X: abnormal_quaker(X)}.
:- not pacifist(a).
:- pacifist(b).
```

- **Answer sets** (1-to-1 correspondence with **stable extensions**)

```
{ abnormal_quaker(b), … }, …
```

- **Rote learning**

```
abnormal_quaker(X) ← X=b
```

The current ABA framework is a **solution**.

It has a stable extension S s.t. pacifist(a) has an argument in S and pacifist(b) has no argument in S.

EXPERIMENTS

ASP-ABAlearnB

https://doi.org/10.5281/zenodo.13330013

SWI Prolog & **Clingo** (ASP)

Potassco

Learning problem	BK	Ep	En	ASP-ABAlearnB	ILASP
Flies	8	4	2	0.01	0.09
Flies_bird&planes	10	5	2	0.02	0.25
Innocent	15	2	2	0.01	1.84
Nixon_diamond	6	1	1	0.01	unsat
Nixon_diamond_2	15	3	2	0.01	unsat
Tax_law	16	2	2	0.02	0.66
Tax_law_2	17	2	2	0.01	0.92
Acute	96	21	19	0.04	unsat
Autism	5716	189	515	23.43	timeout
Breast-w	6291	241	458	16.32	timeout

MORE IN THE PAPER

- Full **ASP-ABAlearnB** algorithm
- Soundness and Termination of **ASP-ABAlearnB**

assumption "relative to"
may lead to failure

- Enforcing Completeness of ASP-ABAlearnB: **ASP-ABAlearnBE**

If "relative to" prevents finding a
solution, then **E**numerate
(Rote Learn some facts w/o generalising)