The 25th International Symposium on Practical Aspects of Declarative Languages

Linear Algebraic Abduction with Partial Evaluation

Tuan Nguyen[1] (speaker), Katsumi Inoue[1] and Chiaki Sakama[2]

[1] *National Institute of Informatics*, Tokyo, Japan

[2] *Wakayama University*, Wakayama, Japan

{tuannq, inoue}@nii.ac.jp sakama@wakayama-u.ac.jp

January 17th, 2023

Outline

Outline

1. Overview and Preliminaries
2. Linear Algebraic Computation of Abduction
3. Partial evaluation
4. Experimental Results
5. Conclusion

Overview and Preliminaries

Abductive reasoning (explanation):
inference to the best explanation starting from a set of observations.

$$\frac{\begin{array}{r} P \Rightarrow Q \\ Q \end{array}}{P}$$

Applications: *Model-based diagnosis*, *belief revision*, *automated reasoning*, ...

Overview and Preliminaries

Definition (Propositional Horn Clause Abduction Problem (PHCAP))

A PHCAP can be modeled as a quadruple $\langle \mathscr{L}, \mathbb{H}, \mathbb{O}, \mathrm{P} \rangle$. Where:

- $\mathscr{L}$ is the set of all propositional variables.
- $\mathbb{H}$ is the set of all hypotheses.
- $\mathbb{O}$ is the set of observations.
- P is a logic program (set of Horn clauses).

- Goal: find the set of minimal explanations $\mathbb{E}$ that satisfies:

Definition (Explanation of PHCAP)

- A set $E \subseteq \mathbb{H}$ is an *explanation* of a PHCAP $\langle \mathscr{L}, \mathbb{H}, \mathbb{O}, \mathrm{P} \rangle$ if $\mathrm{P} \cup E \models \mathbb{O}$ and $\mathrm{P} \cup E$ is consistent.
- An explanation E of $\mathbb{O}$ is *minimal* if there is no explanation E' of $\mathbb{O}$ such that $E' \subset E$.

Overview and Preliminaries

- *Example 1*: An example of PHCAP

$$\mathscr{L} = \{p,\ q,\ r,\ s,\ h_1,\ h_2,\ h_3\},$$
$$\mathbb{H} = \{h_1,\ h_2,\ h_3\},$$
$$\mathbb{O} = \{p\},$$

$$\mathsf{P} = \{\ p \leftarrow q \wedge r,\quad q \leftarrow h_1 \vee s,\quad r \leftarrow s \vee h_2,\quad s \leftarrow h_3\ \}$$

Set of minimal explanations: $\mathbb{E} = \{\ \{h_1,\ h_3\},\ \{h_2,\ h_3\}\ \}$

- Deciding if there is a solution of a PHCAP is NP-complete [1], [2].

[1] Selman and Levesque, "Abductive and Default Reasoning: A Computational Core", 1990.

[2] Eiter and Gottlob, "The complexity of logic-based abduction", 1995.

Overview and Preliminaries

- In this work, we focus on PHCAP with P is an *acyclic program* [3].
- For convenience, P is partitioned into $\mathrm{P}_{And} \cup \mathrm{P}_{Or}$ where:
  - P_{And} is a set of *And*-rule (including facts) and
  - P_{Or} is a set of *Or*-rule.

$$\text{\textit{And}-rule} \qquad h \leftarrow b_1 \wedge \cdots \wedge b_m \quad (m \geq 0)$$

$$\text{\textit{Or}-rule} \qquad h \leftarrow b_1 \vee \cdots \vee b_n \quad (n > 1)$$

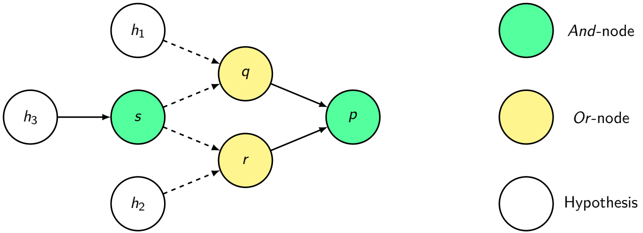
- Standardized program: is a definite program such that there is **no duplicate head atom**.

[3] Apt and Bezem, "Acyclic Programs", 1991.

Overview and Preliminaries

- *Example 1* (continue ...): *And-Or*-graph of a standardized program

$\mathscr{L} = \{p,\ q,\ r,\ s,\ h_1,\ h_2,\ h_3\}$,
$\mathbb{H} = \{h_1,\ h_2,\ h_3\}$,
$\mathbb{O} = \{p\}$,
$\mathrm{P} = \{\ p \leftarrow q \wedge r,$
$q \leftarrow h_1 \vee s,$
$r \leftarrow s \vee h_2,$
$s \leftarrow h_3\ \}$



Outline

Linear Algebraic Computation of Abduction

Definition (Program matrix of PHCAP [4])

Let P be a standardized program and $\mathscr{L} = \{p_1, \ldots, p_n\}$. Then P is represented by a matrix $M_P \in \mathbb{R}^{n \times n}$ such that for each element a_{ij} $(1 \leq i, j \leq n)$ in M_P,

1. $a_{ij_k} = \frac{1}{m}$ $(1 \leq k \leq m;\ 1 \leq i, j_k \leq n)$ if $p_i \leftarrow p_{j_1} \wedge \cdots \wedge p_{j_m}$ (*And*-rule) is in P;
2. $a_{ij_k} = 1$ $(1 \leq k \leq l;\ 1 \leq i, j_k \leq n)$ if $p_i \leftarrow p_{j_1} \vee \cdots \vee p_{j_l}$ (*Or*-rule) is in P;
3. $a_{ii} = 1$ if $p_i \leftarrow$ (fact) is in P or $p_i \in \mathbb{H}$ (abducible);
4. $a_{ij} = 0$, otherwise.

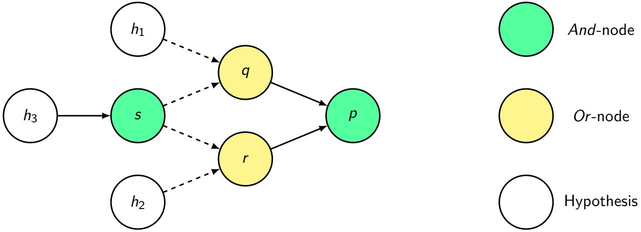
- Any Horn program can be transformed into a standardized program in linear time.
- Horn program $\xrightarrow{\text{standardization}}$ standardized program $\xrightarrow{\text{tensorization}}$ program matrix M_P.

[4] Sakama, Inoue, and Sato, "Linear Algebraic Characterization of Logic Programs", 2017

Linear Algebraic Computation of Abduction

Example 1 (continue ...):

$\mathscr{L} = \{p,\ q,\ r,\ s,\ h_1,\ h_2,\ h_3\}$,
$\mathbb{H} = \{h_1,\ h_2,\ h_3\}$,
$\mathbb{O} = \{p\}$,
$\mathrm{P} = \{\ p \leftarrow q \wedge r,$
$q \leftarrow h_1 \vee s,$
$r \leftarrow s \vee h_2,$
$s \leftarrow h_3\ \}$



	p	q	r	s	h_1	h_2	h_3
p		1/2	1/2				
q				1	1		
r				1		1	
s							1
h_1					1		
h_2						1	
h_3							1

Linear Algebraic Computation of Abduction

Definition (Abductive matrix of PHCAP)

Suppose a PHCAP has P with its *program matrix* M_P.
The *abductive matrix* of P is the transpose of M_P represented as $M_P{}^T$.

Example 1 (continue...): $\mathscr{L} = \{p, q, r, s, h_1, h_2, h_3\}$, $\mathbb{H} = \{h_1, h_2, h_3\}$,
$\mathbb{O} = \{p\}$, $\mathrm{P} = \{p \leftarrow q \wedge r,\ q \leftarrow h_1 \vee s,\ r \leftarrow s \vee h_2,\ s \leftarrow h_3\}$.

$M_P =$

	p	q	r	s	h_1	h_2	h_3
p		1/2	1/2				
q				1	1		
r				1		1	
s							1
h_1					1		
h_2						1	
h_3							1

$M_P{}^T =$

	p	q	r	s	h_1	h_2	h_3
p							
q	1/2						
r	1/2						
s		1	1				
h_1		1			1		
h_2			1			1	
h_3				1			1

Linear Algebraic Computation of Abduction

- Every subset of $\mathscr{L} = \{p,\ q,\ r,\ s,\ h_1,\ h_2,\ h_3\}$ can be represented by a vector.

$$\begin{matrix} p \\ q \\ r \\ s \\ h_1 \\ h_2 \\ h_3 \end{matrix} \begin{pmatrix} \\ \\ \\ \\ 1 \\ 1 \\ 1 \end{pmatrix} \leftrightarrow \mathbb{H} = \{h_1,\ h_2,\ h_3\}$$

Vector of hypotheses

$$\begin{matrix} p \\ q \\ r \\ s \\ h_1 \\ h_2 \\ h_3 \end{matrix} \begin{pmatrix} 1 \\ \\ \\ \\ \\ \\ \\ \end{pmatrix} \leftrightarrow \mathbb{O} = \{p\}$$

Observation vector

- Linear algebraic computation is a set of transformations converting observation vector $\mathbb{O}$ into a vector representing a subset of $\mathbb{H}$. Each transformation step is an 1-step abduction.
- We refer to the vector representing explanations as explanation vector. An *explanation vector* v *reaches an answer* E if $v \subseteq \mathbb{H}$.

Linear Algebraic Computation of Abduction

- If the explanation vector v does not contain head of any *Or*-rule, the abduction step is realized by *matrix multiplication* ${M_P}^T \times v$.

$$
\begin{array}{c} \\ p \\ q \\ r \\ s \\ h_1 \\ h_2 \\ h_3 \end{array}
\begin{array}{c}
\begin{array}{ccccccc} p & q & r & s & h_1 & h_2 & h_3 \end{array} \\
\begin{pmatrix}
 & & & & & & \\
1/2 & & & & & & \\
1/2 & & & & & & \\
 & 1 & 1 & & & & \\
 & 1 & & & 1 & & \\
 & & 1 & & & 1 & \\
 & & & 1 & & & 1
\end{pmatrix}
\end{array}
\times
\begin{array}{c} p \\ q \\ r \\ s \\ h_1 \\ h_2 \\ h_3 \end{array}
\begin{pmatrix} 1 \\ \\ \\ \\ \\ \\ \\ \end{pmatrix}
=
\begin{array}{c} p \\ q \\ r \\ s \\ h_1 \\ h_2 \\ h_3 \end{array}
\begin{pmatrix} \\ 1/2 \\ 1/2 \\ \\ \\ \\ \\ \end{pmatrix}
$$

To explain p, we have to explain both q and r.

- Initial condition: $\sum_{i=1}^{n} v[i] = 1$. A vector is *unexplainable* if $\sum_{i=1}^{n} v[i] < 1$.

Linear Algebraic Computation of Abduction

- If the correspondent vector contains head of any *Or*-rule, the abduction step is realized by *solving a Minimal Hitting Sets (MHS) problem* [5].

$$
\begin{array}{c} \\ p \\ q \\ r \\ s \\ h_1 \\ h_2 \\ h_3 \end{array}
\begin{array}{c}
\begin{array}{ccccccc} p & q & r & s & h_1 & h_2 & h_3 \end{array} \\
\begin{pmatrix}
 & & & & & & \\
1/2 & & & & & & \\
1/2 & & & & & & \\
 & 1 & 1 & & & & \\
 & 1 & & & 1 & & \\
 & & 1 & & & 1 & \\
 & & & 1 & & & 1
\end{pmatrix}
\end{array},
\begin{array}{c} p \\ q \\ r \\ s \\ h_1 \\ h_2 \\ h_3 \end{array}
\begin{pmatrix} \\ 1/2 \\ 1/2 \\ \\ \\ \\ \\ \end{pmatrix}
\xrightarrow{\text{solving MHS}}
\begin{array}{c} p \\ q \\ r \\ s \\ h_1 \\ h_2 \\ h_3 \end{array}
\begin{pmatrix} \\ \\ \\ 1 \\ 1/2 \\ 1/2 \\ \\ \end{pmatrix}
$$

To explain q and r, we have 2 *Or*-rules: $q \leftarrow h_1 \vee s$, $r \leftarrow s \vee h_2$.
Solving a MHS problem: $\{\{h_1, s\}, \{s, h_2\}\}$. Answer: $\{\{s\}, \{h_1, h_2\}\}$.
To explain q and r, we either need to explain s or to explain both h_1 and h_2.

[5] Gainer-Dewar and Vera-Licona, "The minimal hitting set generation problem: algorithms and computation", 2017

Linear Algebraic Computation of Abduction

Definition (*Or*-computable and *And*-computable)

1. A vector v is *Or*-computable iff $v \cap head(\mathrm{P}_{Or}) \neq \emptyset$.
2. A matrix M is *Or*-computable iff $\exists v \in M$, v is *Or*-computable.
3. A vector v is *And*-computable iff v is not *Or*-computable.
4. A matrix M is *And*-computable iff $\forall v \in M$, v is not *Or*-computable.

- For *And*-computable vector/matrix, we can compute the explanations by performing matrix multiplication.
- For *Or*-computable vector/matrix, we can find the explanations by enumerating MHSs.

Outline

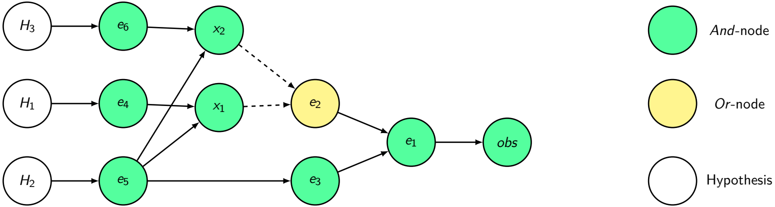
Partial evaluation

Example 2: Consider a program:

$$\mathscr{L} = \{obs,\ e_1,\ e_2,\ e_3,\ e_4,\ e_5,\ e_6,\ H_1,\ H_2,\ H_3\},$$
$$\mathbb{H} = \{H_1,\ H_2,\ H_3\},$$
$$\mathbb{O} = \{obs\},$$

$$P = \{obs \leftarrow e_1,\ e_1 \leftarrow e_2 \wedge e_3,\ e_2 \leftarrow e_4 \wedge e_5,\ e_2 \leftarrow e_5 \wedge e_6,\ e_3 \leftarrow e_5,\ e_4 \leftarrow H_1,\ e_5 \leftarrow H_2,\ e_6 \leftarrow H_3\}.$$

$$P' = \{obs \leftarrow e_1,\ e_1 \leftarrow e_2 \wedge e_3,\ e_2 \leftarrow x_1 \vee x_2,\ e_3 \leftarrow e_5,\ e_4 \leftarrow H_1,\ e_5 \leftarrow H_2,\ e_6 \leftarrow H_3,\ x_1 \leftarrow e_4 \wedge e_5,\ x_2 \leftarrow e_5 \wedge e_6\}.$$



Partial evaluation

Example 2 (continue...):

$$
M_P^T = \begin{array}{c} \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ H_1 \\ H_2 \\ H_3 \\ obs \\ x_1 \\ x_2 \end{array}
\begin{array}{cccccccccccc}
e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & H_1 & H_2 & H_3 & obs & x_1 & x_2 \\
 & & & & & & & & & 1.00 & & \\
0.50 & & & & & & & & & & & \\
0.50 & & & & & & & & & & & \\
 & & & & & & & & & & 0.50 & \\
 & & 1.00 & & & & & & & & 0.50 & 0.50 \\
 & & & & & & & & & & & 0.50 \\
 & & & 1.00 & & & 1.00 & & & & & \\
 & & & & 1.00 & & & 1.00 & & & & \\
 & & & & & 1.00 & & & 1.00 & & & \\
 & & & & & & & & & & & \\
 & 1.00 & & & & & & & & & & \\
 & 1.00 & & & & & & & & & &
\end{array}
$$

Partial evaluation

Example 2 (continue...):

- Iteration 1:

\- $M^{(1)} = \theta(M_P^T \times M^{(0)})$, where $M^{(0)} = \mathbb{O}$:

$M^{(1)}$:

	0
e_1	1.00
e_2	
e_3	
e_4	
e_5	
e_6	
H_1	
H_2	
H_3	
obs	
x_1	
x_2	

=

M_P^T:

	e_1	e_2	e_3	e_4	e_5	e_6	H_1	H_2	H_3	obs	x_1	x_2
e_1										1.00		
e_2	0.50											
e_3	0.50											
e_4											0.50	
e_5			1.00								0.50	0.50
e_6												0.50
H_1				1.00			1.00					
H_2					1.00			1.00				
H_3						1.00			1.00			
obs												
x_1		1.00										
x_2		1.00										

×

$M^{(0)}$:

	0
e_1	
e_2	
e_3	
e_4	
e_5	
e_6	
H_1	
H_2	
H_3	
obs	1.00
x_1	
x_2	

(*) Vector/matrix can be represented in sparse format : Coordinate (COO) / Compressed Sparse Row (CSR) / Compressed Sparse Column (CSC).

Partial evaluation

Example 2 (continue...):

- Iteration 2:

- $M^{(2)} = \theta(M_P^T \times M^{(1)})$

$M^{(2)}$:

	0
e_1	
e_2	0.50
e_3	0.50
e_4	
e_5	
e_6	
H_1	
H_2	
H_3	
obs	
x_1	
x_2	

=

	e_1	e_2	e_3	e_4	e_5	e_6	H_1	H_2	H_3	obs	x_1	x_2
e_1										1.00		
e_2	0.50											
e_3	0.50											
e_4											0.50	
e_5			1.00								0.50	0.50
e_6												0.50
H_1				1.00			1.00					
H_2					1.00			1.00				
H_3						1.00			1.00			
obs												
x_1		1.00										
x_2		1.00										

×

	0
e_1	1.00
e_2	
e_3	
e_4	
e_5	
e_6	
H_1	
H_2	
H_3	
obs	
x_1	
x_2	

- Solving MHS: $\{\ \{x_1,\ x_2\},\ \{e_3\}\ \}$.

MHS solutions: $\{\ \{e_3,\ x_1\},\ \{e_3,\ x_2\}\ \} = M^{(3)}$.

Partial evaluation

Example 2 (continue...):

- Iteration 3:

- $M^{(4)} = \theta(M_P^T \times M^{(3)})$

	0	1
e_1		
e_2		
e_3		
e_4	0.25	
e_5	0.75	0.75
e_6		0.25
H_1		
H_2		
H_3		
obs		
x_1		
x_2		

=

	e_1	e_2	e_3	e_4	e_5	e_6	H_1	H_2	H_3	obs	x_1	x_2
e_1										1.00		
e_2	0.50											
e_3	0.50											
e_4											0.50	
e_5			1.00								0.50	0.50
e_6												0.50
H_1				1.00			1.00					
H_2					1.00			1.00				
H_3						1.00			1.00			
obs												
x_1		1.00										
x_2		1.00										

×

	0	1
e_1		
e_2		
e_3	0.50	0.50
e_4		
e_5		
e_6		
H_1		
H_2		
H_3		
obs		
x_1	0.50	
x_2		0.50

Partial evaluation

Example 2 (continue...):

- Iteration 4:

- $M^{(4)} = \theta(M_P^T \times M^{(3)})$

	0	1
e_1		
e_2		
e_3		
e_4		
e_5		
e_6		
H_1	0.25	
H_2	0.75	0.75
H_3		0.25
obs		
x_1		
x_2		

=

	e_1	e_2	e_3	e_4	e_5	e_6	H_1	H_2	H_3	obs	x_1	x_2
e_1										1.00		
e_2	0.50											
e_3	0.50											
e_4											0.50	
e_5			1.00								0.50	0.50
e_6												0.50
H_1				1.00			1.00					
H_2					1.00			1.00				
H_3						1.00			1.00			
obs												
x_1		1.00										
x_2		1.00										

×

	0	1
e_1		
e_2		
e_3		
e_4	0.25	
e_5	0.75	0.75
e_6		0.25
H_1		
H_2		
H_3		
obs		
x_1		
x_2		

- The algorithm stops. Found minimal explanations: $\{ \{H_1, H_2\}, \{H_2, H_3\} \}$.

Partial evaluation

Definition (**Reduct abductive matrix**)

We can obtain a *reduct abductive matrix* $M_P(\mathrm{P}'^{r}_{And})^T$ from the abductive matrix $M_P{}^T$ by:

1. Removing all columns *w.r.t.* *Or*-rules in P_{Or}.
2. Setting 1 at the diagonal corresponding to all atoms which are heads of *Or*-rules.

Consider the PHCAP in *Example 2*:

$$M_P(P'^{r}_{And})^T =$$

	e_1	e_2	e_3	e_4	e_5	e_6	H_1	H_2	H_3	obs	x_1	x_2
e_1										1.00		
e_2	0.50	1.00										
e_3	0.50											
e_4											0.50	
e_5			1.00								0.50	0.50
e_6												0.50
H_1				1.00			1.00					
H_2					1.00			1.00				
H_3						1.00			1.00			
obs												
x_1												
x_2												

Partial evaluation

Definition (**Partial evaluation in abduction**)

Let a PHCAP $\langle \mathscr{L}, \mathbb{H}, \mathbb{O}, \mathrm{P} \rangle$ where P is a standardized program.
For any And$-$*rule* $r = (h \leftarrow b_1 \wedge \cdots \wedge b_m)$ in P:

- if $body(r)$ contains an atom b_i $(1 \leq i \leq m)$ which is not the head of any rule in P, then remove r.
- otherwise, for each atom $b_i \in body(r)$ $(i = 1, \ldots, m)$, if there is an *And*-rule $b_i \leftarrow B_i$ in P (where B_i is a conjunction of atoms), then replace each b_i in $body(r)$ by the conjunction B_i.

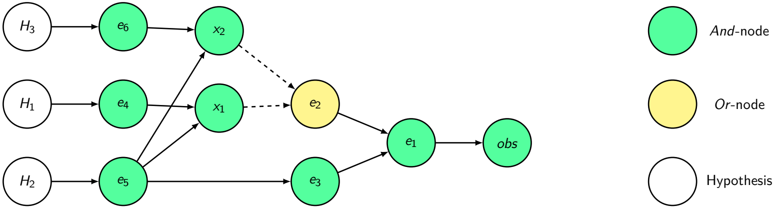
The resulting rule is denoted by unfold(r). Define

$$\mathrm{peval}(P) = \bigcup_{r \in P_{And}} \mathrm{unfold}(r).$$

peval(P) is called *partial evaluation* of P.

Partial evaluation

Example 3: Consider a similar program in *Example 2*:
$\mathscr{L} = \{obs, e_1, e_2, e_3, e_4, e_5, e_6, H_1, H_2, H_3\}$, $\mathbb{H} = \{H_1, H_2, H_3\}$, $\mathbb{O} = \{obs\}$,
$P = \{obs \leftarrow e_1,\ e_1 \leftarrow e_2 \wedge e_3,\ e_2 \leftarrow e_4 \wedge e_5,\ e_2 \leftarrow e_5 \wedge e_6,\ e_3 \leftarrow e_5,\ e_4 \leftarrow H_1,\ e_5 \leftarrow H_2,\ e_6 \leftarrow H_3\}$.
Standardized program $P' = \{obs \leftarrow e_1,\ e_1 \leftarrow e_2 \wedge e_3,\ e_2 \leftarrow x_1 \vee x_2,\ e_3 \leftarrow e_5,\ e_4 \leftarrow H_1,\ e_5 \leftarrow H_2,\ e_6 \leftarrow H_3,\ x_1 \leftarrow e_4 \wedge e_5,\ x_2 \leftarrow e_5 \wedge e_6\}$.



Partial evaluation

Example 3 (continue...):

- Let $P' = \{r_1, ..., r_9\}$ where:

$r_1 = (obs \leftarrow e_1)$,
$r_2 = (e_1 \leftarrow e_2 \wedge e_3)$,
$r_3 = (e_2 \leftarrow x_1 \vee x_2)$,
$r_4 = (x_1 \leftarrow e_4 \wedge e_5)$,
$r_5 = (x_2 \leftarrow e_5 \wedge e_6)$,
$r_6 = (e_3 \leftarrow e_5)$,
$r_7 = (e_4 \leftarrow H_1)$,
$r_8 = (e_5 \leftarrow H_2)$,
$r_9 = (e_6 \leftarrow H_3)$.

- Unfolding rules of P' becomes:

$\text{unfold}(r_1) = (obs \leftarrow e_2 \wedge e_3)$,
$\text{unfold}(r_2) = (e_1 \leftarrow e_2 \wedge e_5)$,
$\text{unfold}(r_3) = r_3$,
$\text{unfold}(r_4) = (x_1 \leftarrow H_1 \wedge H_2)$,
$\text{unfold}(r_5) = (x_2 \leftarrow H_2 \wedge H_3)$,
$\text{unfold}(r_6) = (e_3 \leftarrow H_2)$,
$\text{unfold}(r_7) = r_7$,
$\text{unfold}(r_8) = r_8$,
$\text{unfold}(r_9) = r_9$.

- Then $\text{peval}(P')$ consists of:

$obs \leftarrow e_2 \wedge e_3$,
$e_1 \leftarrow e_2 \wedge e_5$,
$e_2 \leftarrow x_1 \vee x_2$,
$x_1 \leftarrow H_1 \wedge H_2$,
$x_2 \leftarrow H_2 \wedge H_3$,
$e_3 \leftarrow H_2$,
$e_4 \leftarrow H_1$,
$e_5 \leftarrow H_2$,
$e_6 \leftarrow H_3$.

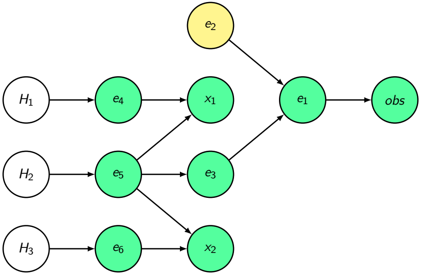
Partial evaluation

- peval(P') can be obtained by computing the power of the *reduct abductive matrix*: $\left(M_P(P'^r_{And})^T\right)^2$, $\left(M_P(P'^r_{And})^T\right)^4$, ... $\left(M_P(P'^r_{And})^T\right)^{2^k}$ where k is the number of peval steps.

Example 3 (continue...):

$M_P(P'^r_{And})^T =$

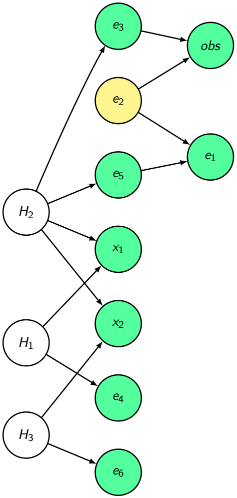
	e_1	e_2	e_3	e_4	e_5	e_6	H_1	H_2	H_3	obs	x_1	x_2
e_1										1.00		
e_2	0.50	1.00										
e_3	0.50											
e_4											0.50	
e_5			1.00								0.50	0.50
e_6												0.50
H_1				1.00			1.00					
H_2					1.00			1.00				
H_3						1.00			1.00			
obs												
x_1												
x_2												



Partial evaluation

$$\left(M_P({P'}^{r}_{And})^T\right)^2 =$$

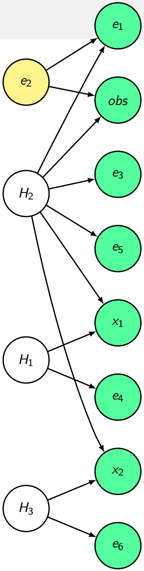
	e_1	e_2	e_3	e_4	e_5	e_6	H_1	H_2	H_3	obs	x_1	x_2
e_1												
e_2	0.50	1.00								0.50		
e_3										0.50		
e_4												
e_5	0.50											
e_6												
H_1				1.00			1.00				0.50	
H_2			1.00		1.00			1.00			0.50	0.50
H_3						1.00			1.00			0.50
obs												
x_1												
x_2												



Partial evaluation

$$\left(\left(M_P(P'^{r}_{And})^T\right)^2\right)^2 =$$

	e_1	e_2	e_3	e_4	e_5	e_6	H_1	H_2	H_3	obs	x_1	x_2
e_1												
e_2	0.50	1.00								0.50		
e_3												
e_4												
e_5												
e_6												
H_1				1.00			1.00				0.50	
H_2	0.50		1.00		1.00			1.00		0.50	0.50	0.50
H_3						1.00			1.00			0.50
obs												
x_1												
x_2												



- Here, we reach a fixpoint at $k = 2$. We refer to this "stable" matrix as peval(P) and take it to solve the PHCAP.

Partial evaluation

Example 3 (continue...):

- Iteration 1:

- $M^{(1)} = \theta(\text{peval}(P) \times M^{(0)})$, where $M^{(0)} = \mathbb{O}$

$M^{(1)}$ =

	0
e_1	
e_2	0.50
e_3	
e_4	
e_5	
e_6	
H_1	
H_2	0.50
H_3	
obs	
x_1	
x_2	

=

	e_1	e_2	e_3	e_4	e_5	e_6	H_1	H_2	H_3	obs	x_1	x_2
e_1												
e_2	0.50	1.00								0.50		
e_3												
e_4												
e_5												
e_6												
H_1				1.00			1.00				0.50	
H_2	0.50		1.00		1.00			1.00		0.50	0.50	0.50
H_3						1.00			1.00			0.50
obs												
x_1												
x_2												

×

	0
e_1	
e_2	
e_3	
e_4	
e_5	
e_6	
H_1	
H_2	
H_3	
obs	1.00
x_1	
x_2	

- Solving MHS problem: $\{\ \{x_1,\ x_2\},\ \{H_2\}\ \}$.

MHS solutions: $\{\ \{H_2,\ x_1\},\ \{H_2,\ x_2\}\ \} = M^{(2)}$.

Partial evaluation

Example 3 (continue...):

- Iteration 2:

- $M^{(3)} = \theta(\text{peval}(P) \times M^{(2)})$

$M^{(3)}$:

	0	1
e_1		
e_2		
e_3		
e_4		
e_5		
e_6		
H_1		0.25
H_2	0.75	0.75
H_3	0.25	
obs		
x_1		
x_2		

=

	e_1	e_2	e_3	e_4	e_5	e_6	H_1	H_2	H_3	obs	x_1	x_2
e_1												
e_2	0.50	1.00								0.50		
e_3												
e_4												
e_5												
e_6												
H_1				1.00			1.00				0.50	
H_2	0.50		1.00		1.00			1.00		0.50	0.50	0.50
H_3						1.00			1.00			0.50
obs												
x_1												
x_2												

×

	0	1
e_1		
e_2		
e_3		
e_4		
e_5		
e_6		
H_1		
H_2	0.50	0.50
H_3		
obs		
x_1		0.50
x_2	0.50	

- The algorithm stops. Found minimal explanations: $\{\ \{H_1,\ H_2\},\ \{H_2,\ H_3\}\ \}$.

Partial evaluation

- Partial evaluation is repeatedly performed as:

$$\text{peval}^0(P) = P \quad \text{and} \quad \text{peval}^k(P) = \text{peval}(\text{peval}^{k-1}(P)) \;\; (k \geq 1).$$

- It is realized as computing the power of the reduct abductive matrix:

$$\left(M_P(P''^{r}_{And})^T\right)^2, \left(M_P(P''^{r}_{And})^T\right)^4, \ldots \left(M_P(P''^{r}_{And})^T\right)^{2^k} \quad (k \geq 1)$$

Partial evaluation

- Partial evaluation is repeatedly performed as:

$$\text{peval}^0(P) = P \quad \text{and} \quad \text{peval}^k(P) = \text{peval}(\text{peval}^{k-1}(P)) \;\; (k \geq 1).$$

- It is realized as computing the power of the reduct abductive matrix:

$$\left(M_P(P^{r}_{And})^T\right)^2,\; \left(M_P(P^{r}_{And})^T\right)^4,\; \ldots \left(M_P(P^{r}_{And})^T\right)^{2^k} \;\; (k \geq 1)$$

- Partial evaluation has a fixpoint (the proof is presented in our paper).

Partial evaluation

- Partial evaluation is repeatedly performed as:

$$\text{peval}^0(P) = P \quad \text{and} \quad \text{peval}^k(P) = \text{peval}(\text{peval}^{k-1}(P)) \;\; (k \geq 1).$$

- It is realized as computing the power of the reduct abductive matrix:

$$\left(M_P(P^{\prime r}_{And})^T\right)^2, \left(M_P(P^{\prime r}_{And})^T\right)^4, \ldots \left(M_P(P^{\prime r}_{And})^T\right)^{2^k} \;\; (k \geq 1)$$

- Partial evaluation has a fixpoint (the proof is presented in our paper).
- The k-step partial evaluation has the effect of realizing 2^k steps of deduction at once. Multiplying an explanation vector and the peval matrix thus realizes exponential speed-up.

Partial evaluation

- Partial evaluation is repeatedly performed as:

$$\text{peval}^0(P) = P \quad \text{and} \quad \text{peval}^k(P) = \text{peval}(\text{peval}^{k-1}(P)) \ \ (k \geq 1).$$

- It is realized as computing the power of the reduct abductive matrix:

$$\left(M_P(P^{\prime r}_{And})^T\right)^2, \ \left(M_P(P^{\prime r}_{And})^T\right)^4, \ \ldots \left(M_P(P^{\prime r}_{And})^T\right)^{2^k} \ \ (k \geq 1)$$

- Partial evaluation has a fixpoint (the proof is presented in our paper).
- The k-step partial evaluation has the effect of realizing 2^k steps of deduction at once. Multiplying an explanation vector and the peval matrix thus realizes exponential speed-up.
- However, computing the power of matrix is costly. We need to verify the positive effect can win the tradeoff.

Outline

Experimental Results

- We experiment on Failure Modes and Effects Analysis (FMEA)-based benchmark datasets by Koitz-Hristov and Wotawa which has been used in [6] and [7].

Dataset	Number of instances	Characteristics
Artificial samples I	166 problems	deeper but narrower graph structure
Artificial samples II	117 problems [8]	deeper and wider graph structure, some problems involve solving a large number of medium-size MHS problems
FMEA samples	213 problems	shallower but wider graph structure, usually involving a few (but) large-size MHS problems

[6] Koitz-Hristov and Wotawa, "Applying algorithm selection to abductive diagnostic reasoning", 2018.

[7] Koitz-Hristov and Wotawa, "Faster Horn diagnosis-a performance comparison of abductive reasoning algorithms", 2020.

[8] Excluded the unresolved problem `phcap_140_5_5_5.atms`

Experimental Results

- We implement our method as two versions: *Dense matrix* and *Sparse matrix* in Python 3.7 (using Numpy and Scipy). Each version we have one with partial evaluation and one without partial evaluation.
- For large-size MHS problems, which have more than 50,000 posible combinations, we use MHS enumerator provided by PySAT [9].
  - All the source code and benchmark datasets in our paper are available on GitHub:

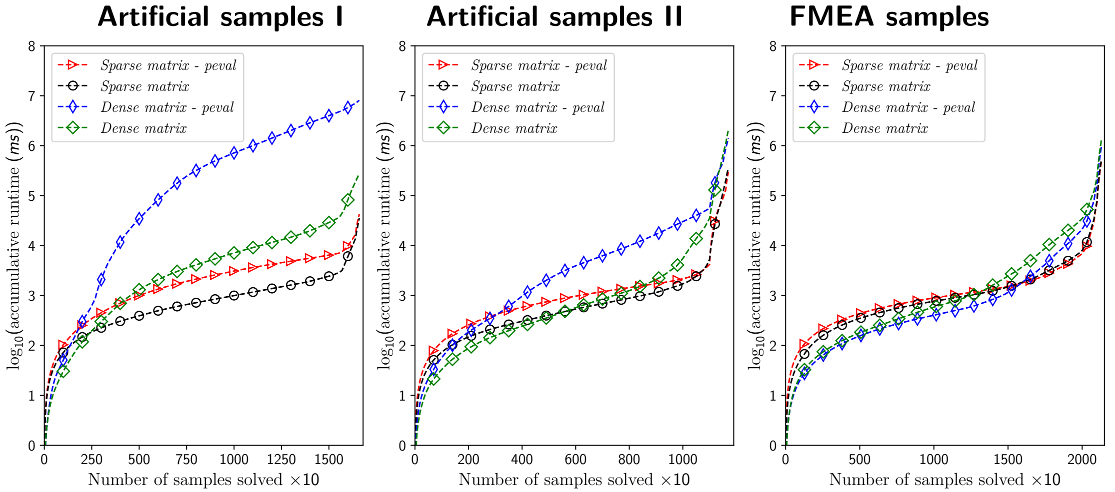
`https://github.com/nqtuan0192/LinearAlgebraicComputationofAbduction`.

- We have demonstrated the performance of linear algebraic approaches in [10].

[9] Ignatiev, Morgado, and Marques-Silva, "PySAT: A Python Toolkit for Prototyping with SAT Oracles", 2018.

[10] Nguyen, Inoue, and Sakama, "Linear algebraic computation of propositional Horn abduction", 2021.

Experimental Results - Original benchmark



Experimental Results - Original benchmark

Table: Detailed execution results for the original benchmark.

Datasets	**Artificial samples I** (166 problems)			**Artificial samples II** (117 problems)			**FMEA samples** (213 problems)		
Algorithms	**#solved / #fastest**	$t + t_p$ **mean / std**	t_{peval} **mean / std**	**#solved / #fastest**	$t + t_p$ **mean / std**	t_{peval} **mean / std**	**#solved / #fastest**	$t + t_p$ **mean / std**	t_{peval} **mean / std**
Sparse matrix - peval	**1,660** / 89	4,243 / 93	514 / 19	**1,170** / 246	**29,438** / 112	124 / 48	**2,130** / 726	**49,481** / 1,214	84 / 4
Sparse matrix	**1,660** / 1,401	**3,527** / 29	- / -	**1,170** / 513	35,844 / 62	- / -	**2,130** / 150	53,553 / 1,254	- / -
Dense matrix - peval	**1,660** / 13	811,841 / 2,227	728,086 / 31,628	**1,170** / 90	140,589 / 1,293	3,599 / 910	**2,130** / 1,007	98,614 / 2,950	25 / 3
Dense matrix	**1,660** / 157	27,569 / 183	- / -	**1,170** / 321	205,279 / 1,866	- / -	**2,130** / 247	131,734 / 3,629	- / -

- Partial evaluation improves much more in **Artificial samples II** and **FMEA samples**.
- We see performance degradation happens in **Artificial samples I** for both dense and sparse methods, especially with the dense method.

Experimental Results - Enhanced benchmark datasets

In this experiment, we enhance the benchmark dataset based on the transitive closure problem:

$$P = \{\ path(X,\ Y) \leftarrow edge(X,\ Y),$$
$$path(X,\ Y) \leftarrow edge(X,\ Z) \wedge path(Z,\ Y)\ \}$$

Experimental Results - Enhanced benchmark datasets

In this experiment, we enhance the benchmark dataset based on the transitive closure problem:

$$P = \{\ path(X,\ Y) \leftarrow edge(X,\ Y),$$
$$path(X,\ Y) \leftarrow edge(X,\ Z) \wedge path(Z,\ Y)\ \}$$

- First, generate a PHCAP based on the transitive closure of a single line graph: $edge(1,\ 2)$, $edge(2,\ 3)$, $edge(3,\ 4)$, $edge(4,\ 5)$, $edge(5,\ 6)$, $edge(6,\ 7)$, $edge(7,\ 8)$, $edge(8,\ 9)$, $edge(9,\ 10)$.

Experimental Results - Enhanced benchmark datasets

In this experiment, we enhance the benchmark dataset based on the transitive closure problem:

$$P = \{\ path(X,\ Y) \leftarrow edge(X,\ Y),$$
$$path(X,\ Y) \leftarrow edge(X,\ Z) \wedge path(Z,\ Y)\ \}$$

- First, generate a PHCAP based on the transitive closure of a single line graph: $edge(1,\ 2)$, $edge(2,\ 3)$, $edge(3,\ 4)$, $edge(4,\ 5)$, $edge(5,\ 6)$, $edge(6,\ 7)$, $edge(7,\ 8)$, $edge(8,\ 9)$, $edge(9,\ 10)$.
- Then we consider the observation to be $path(1,\ 10)$, and look for the explanation of it.

Experimental Results - Enhanced benchmark datasets

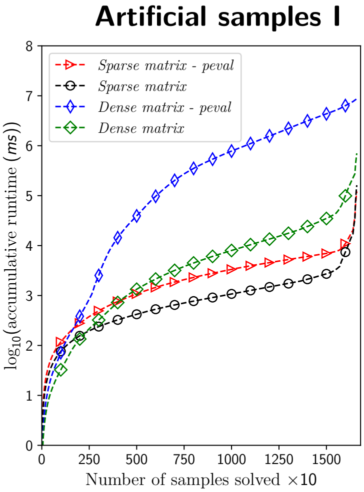
In this experiment, we enhance the benchmark dataset based on the transitive closure problem:

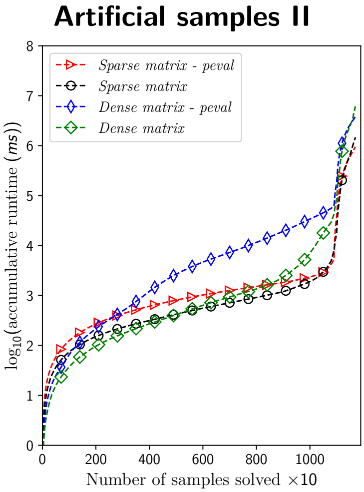
$$P = \{\ path(X,\ Y) \leftarrow edge(X,\ Y),$$
$$path(X,\ Y) \leftarrow edge(X,\ Z) \wedge path(Z,\ Y)\ \}$$

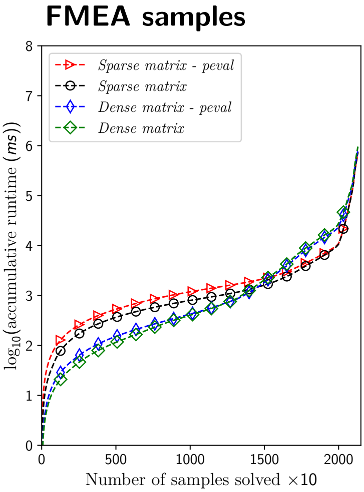
- First, generate a PHCAP based on the transitive closure of a single line graph: $edge(1,\ 2)$, $edge(2,\ 3)$, $edge(3,\ 4)$, $edge(4,\ 5)$, $edge(5,\ 6)$, $edge(6,\ 7)$, $edge(7,\ 8)$, $edge(8,\ 9)$, $edge(9,\ 10)$.
- Then we consider the observation to be $path(1,\ 10)$, and look for the explanation of it.
- Next, for each problem instance of the original benchmark, we enumerate rules of the form $e \leftarrow h$, where h is a hypothesis and e is a propositional variable , and append the atom of the observation of the new PHCAP into this rule with a probability of 20% .

The resulting problem is expected to have the subgraph of And-rules occur more frequently.

Experimental Results - Enhanced benchmark datasets







Experimental Results - Enhanced benchmark datasets

Table: Detailed execution results for the enhanced benchmark datasets.

Datasets	**Artificial samples I** (166 problems)			**Artificial samples II** (117 problems)			**FMEA samples** (213 problems)		
Algorithms	**#solved / #fastest**	$t + t_p$ **mean / std**	t_{peval} **mean / std**	**#solved / #fastest**	$t + t_p$ **mean / std**	t_{peval} **mean / std**	**#solved / #fastest**	$t + t_p$ **mean / std**	t_{peval} **mean / std**
Sparse matrix - peval	**1,660** / 116	**12,140** / 124	545 / 15	**1,170** / 254	**95,079** / 616	138 / 4	**2,130** / 384	**72,776** / 1,103	157 / 5
Sparse matrix	**1,660** / 1,389	16,163 / 209	- / -	**1,170** / 516	147,444 / 1,508	- / -	**2,130** / 553	74,861 / 526	- / -
Dense matrix - peval	**1,660** / 5	869,922 / 2,434	799,965 / 58,500	**1,170** / 77	380,033 / 2,228	4,483 / 688	**2,130** / 436	81,837 / 1,005	103 / 10
Dense matrix	**1,660** / 150	70,365 / 681	- / -	**1,170** / 323	613,422 / 3,651	- / -	**2,130** / 757	95,996 / 1,021	- / -

- With the dataset enhancement, we now see partial evaluation can improve the performance for sparse method significantly in the **Artificial samples I**.
- However, the problem still remains with the dense method.
- The graph structure of the **Artificial samples I** is the cause of the problem. That we take more time in computing the power of the matrix with the dense format.
- It also hightlights the importance of sparse representation.

Outline

Conclusion

Contributions:

1. We have proposed to improve the linear algebraic approach for abduction by employing partial evaluation.
2. Partial evaluation steps can be realized as the power of the reduct abductive matrix in the language of linear algebra.
3. Its significant enhancement in terms of execution time has been demonstrated using artificial benchmarks and real FMEA-based datasets with both dense and sparse representation, especially more with the sparse format.

Conclusion

But why do we need linear algebraic method?

1. It simplifies the core algorithm (easy to understand, easy to implement).
2. It can take the advantages of recent advancements in tensor oriented computing hardwares.
3. It is expected to be better scalability.

Conclusion

Future work:

1. Handling loops and extending the method to work with non-Horn clauses.
2. Employing an effective prediction to know better when to apply partial evaluation and how deep we do unfolding before solving the problem.
3. Moreover, incorporating some efficient pruning techniques or knowing where to zero out in the abductive matrix is also a potential topic.

References I

Apt, Krzysztof R. and Marc Bezem. “Acyclic Programs”. In: *New Generation Computing* 9 (1991), pp. 335–364.

Eiter, Thomas and Georg Gottlob. “The complexity of logic-based abduction”. In: *Journal of the ACM (JACM)* 42.1 (1995), pp. 3–42.

Gainer-Dewar, Andrew and Paola Vera-Licona. “The minimal hitting set generation problem: algorithms and computation”. In: *SIAM Journal on Discrete Mathematics* 31.1 (2017), pp. 63–100.

Ignatiev, Alexey, Antonio Morgado, and Joao Marques-Silva. “PySAT: A Python Toolkit for Prototyping with SAT Oracles”. In: *SAT*. 2018, pp. 428–437.

Koitz-Hristov, Roxane and Franz Wotawa. “Applying algorithm selection to abductive diagnostic reasoning”. In: *Applied Intelligence* 48.11 (2018), pp. 3976–3994.

– .“Faster Horn diagnosis-a performance comparison of abductive reasoning algorithms”. In: *Applied Intelligence* 50.5 (2020), pp. 1558–1572.

References II

Nguyen, Tuan Quoc, Katsumi Inoue, and Chiaki Sakama. “Linear algebraic computation of propositional Horn abduction”. In: *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE. 2021, pp. 240–247. DOI: 10.1109/ICTAI52525.2021.00040.

Sakama, Chiaki, Katsumi Inoue, and Taisuke Sato. “Linear Algebraic Characterization of Logic Programs”. In: *International Conference on Knowledge Science, Engineering and Management*. Springer. 2017, pp. 520–533.

Selman, Bart and Hector J Levesque. “Abductive and Default Reasoning: A Computational Core”. In: *AAAI*. 1990, pp. 343–348.

Thank you for your attention