The 33rd IEEE International Conference on Tools with Artificial Intelligence

Linear Algebraic Computation of Propositional Horn Abduction

Tuan Nguyen[1,2], Katsumi Inoue[2,1] and Chiaki Sakama[3,2]

[1] *The Graduate University for Advanced Studies*, SOKENDAI, Tokyo, Japan

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

[3] *Wakayama University*, Wakayama, Japan

November 1st, 2021

Outline

Outline

Overview

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

$$\begin{array}{l} P \Rightarrow Q \\ \quad\quad Q \\ \hline P \end{array}$$

Overview

- In this paper, we focus on Propositional Horn Clause Abduction Problem (PHCAP) with *acyclic program*[1].

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

Definition

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

- Deciding if there is a solution of a PHCAP is NP-complete[2,3].

[1] Apt and Bezem, "Acyclic Programs", 1991.
[2] Selman and Levesque, "Abductive and Default Reasoning: A Computational Core", 1990.
[3] Eiter and Gottlob, "The complexity of logic-based abduction", 1995.

Outline

Abduction using Matrix Representation

Definition (**Program matrix** [4] **of PHCAP**)

Let P be a standardized program and $B_P = \{p_1, \ldots, p_n\}$. 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 $p_i \in \mathbb{H}$;
4. $a_{ij} = 0$, otherwise.

Example 2: The PHCAP in *Example 1*
$\mathcal{L} = \{p,\ q,\ r,\ s,\ h_1,\ h_2,\ h_3\}$,
$\mathbb{H} = \{h_1,\ h_2,\ h_3\}$, $\mathbb{O} = \{p\}$,
$P = \{p \leftarrow q \wedge r,\ q \leftarrow h_1 \vee s,\ r \leftarrow s \vee h_2,\ s \leftarrow h_3\}$.

	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

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

Abduction using Matrix Representation

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 3: $\mathcal{L} = \{p,\ q,\ r,\ s,\ h_1,\ h_2,\ h_3\}$, $\mathbb{H} = \{h_1,\ h_2,\ h_3\}$, $\mathbb{O} = \{p\}$,
$P = \{p \leftarrow q \wedge r,\ q \leftarrow h_1 \vee s,\ r \leftarrow s \vee h_2,\ s \leftarrow h_3\}$.

$$M_P = \begin{array}{c} \\ p \\ q \\ r \\ s \\ h_1 \\ h_2 \\ h_3 \end{array} \begin{array}{c} \begin{array}{ccccccc} p & q & r & s & h_1 & h_2 & h_3 \end{array} \\ \left(\begin{array}{ccccccc} & 1/2 & 1/2 & & & & \\ & & & 1 & 1 & & \\ & & & 1 & & 1 & \\ & & & & & & 1 \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & & & 1 \end{array}\right) \end{array},\quad {M_P}^T = \begin{array}{c} \\ p \\ q \\ r \\ s \\ h_1 \\ h_2 \\ h_3 \end{array} \begin{array}{c} \begin{array}{ccccccc} p & q & r & s & h_1 & h_2 & h_3 \end{array} \\ \left(\begin{array}{ccccccc} & & & & & & \\ 1/2 & & & & & & \\ 1/2 & & & & & & \\ & 1 & 1 & & & & \\ & 1 & & & 1 & & \\ & & 1 & & & 1 & \\ & & & 1 & & & 1 \end{array}\right) \end{array}$$

Abduction using Matrix Representation

- *Interpretation vector* v is a vector which represents the truth value of propositions.

$$v = \begin{pmatrix} \overset{p}{1} & \overset{q}{0} & \overset{r}{0} & \overset{s}{0} & \overset{h_1}{0} & \overset{h_2}{0} & \overset{h_3}{0} \end{pmatrix}^T \text{ or we say: } v = \{p\}$$

- Initial condition: $\sum_{i=1}^{n} v[i] = 1$
- θ-thresholding: limiting the value of every element by 1.
- An interpretation vector is called insufficient to be an explanation if $\sum_{i=1}^{n} v[i] < 1$.
- These above definitions and conditions can also be applied to multiple vectors that form a interpretation matrix.

Abduction using Matrix Representation

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

1. A vector v is *Or*-computable iff $v \cap head(\mathbb{T}_{Or}) \neq \emptyset$.
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 explanation by performing matrix multiplication.
- For *Or*-computable vector/matrix, we can find the explanation by enumerating Minimal Hitting Sets (MHS) [5].

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

Abduction using Matrix Representation

Example 4:

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

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

$$M^{(0)} = \mathbb{O} = \begin{array}{c} \begin{array}{ccccccc} p & q & r & s & h_1 & h_2 & h_3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$M^{(1)} = M_P{}^T \cdot M^{(0)} = \begin{array}{c} \begin{array}{ccccccc} p & q & r & s & h_1 & h_2 & h_3 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$M^{(1)} = \{\{q, r\}\}$$

$$\mathbb{S}_{(M_0^{(1)},\ P_{Or})} = \{\{h_1,\ s\},\ \{s,\ h_2\}\}$$

$$\mathbf{MHS}(\mathbb{S}_{(M_0^{(1)},\ P_{Or})}) = \{\{s\},\ \{h_1,\ h_2\}\}\ (= M^{(2)})$$

$$M^{(2)} = \begin{array}{c} \begin{array}{ccccccc} p & q & r & s & h_1 & h_2 & h_3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix} \end{array}$$

$$M^{(3)} = M_P{}^T \cdot M^{(2)} = \begin{array}{c} \begin{array}{ccccccc} p & q & r & s & h_1 & h_2 & h_3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix} \end{array}$$

$$\mathbb{E} = \{\{h_3\}, \{h_1, h_2\}\}$$

Abduction using Matrix Representation

Definition

1-step abduction for P_{And}:

$$M^{(t+1)} = M_P(P_{And})^T \cdot M^{(t)} \qquad (1)$$

Definition

1-step abduction for P_{Or}:

$$M^{(t+1)} = \bigcup_{\forall v \in M^{(t)}} \; \bigcup_{\forall s \in \mathbf{MHS}(\mathbb{S}_{(v,\ P_{Or})})} \Big((v \setminus head(P_{Or})) \cup s \Big) \qquad (2)$$

where: $\mathbb{S}_{(v,\ P_{Or})} = \{body(r_1), body(r_2), \dots, body(r_k)\}$ such that $v \cap head(P_{Or}) = \{head(r_1), head(r_2), \dots, head(r_k)\}$.

Algorithm 1 Explanations finding in a vector space

Input: PHCAP consists of a tuple $\langle \mathcal{L}, \mathbb{H}, \mathbb{O}, P \rangle$
Output: Set of explanations $\mathbb{E}$

Create an abductive matrix ${M_P}^T$ from P
Initialize the observation matrix M from $\mathbb{O}$
$\mathbb{E} = \emptyset$
while True do
  $M' = {M_P}^T \cdot M$
  $M' = \mathbf{consistent}(M')$
  $v\_sum = sum_{col}(M') < 1 - \epsilon$
  $M' = M'[v\_sum = \mathbf{False}]$
  if $M' = M$ **or** $M' = \emptyset$ **then**
    $v\_ans = \theta(M + \mathbb{H}) \leq \theta(\mathbb{H})$
    $\mathbb{E} = \mathbb{E} \cup M[v\_ans = \mathbf{True}]$
    return minimal$(\mathbb{E})$
  do
    $v\_ans = \theta(M' + \mathbb{H}) \leq \theta(\mathbb{H})$
    $\mathbb{E} = \mathbb{E} \cup M'[v\_ans = \mathbf{True}]$
    $M' = M'[v\_ans = \mathbf{False}]$
    $M = M \cup M'[\mathbf{not}\ Or\text{-computable}]$
    $M' = M'[Or\text{-computable}]$
    $M' = \bigcup_{\forall v \in M'} \bigcup_{\forall s \in \mathbf{MHS}(\mathbb{S}_{(v,\ P_{Or})})} \Big((v \setminus head(P_{Or})) \cup s \Big)$
    $M' = \mathbf{consistent}(M')$
  while $M' \neq \emptyset$

Outline

Experimental Results

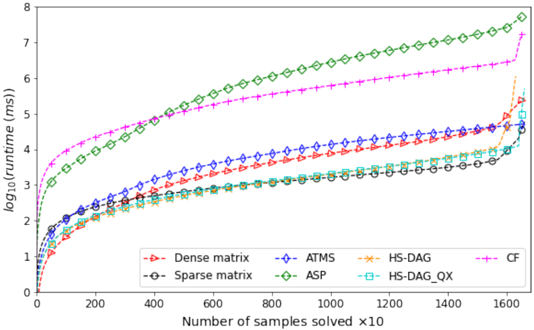
- We experiment on the benchmark dataset by Koitz-Hristov and Wotawa [6], [7].
  - Artificial samples I: artificial, deeper graph structure.
  - Artificial samples II: artificial, deeper graph structure, some problem involves solving a large number of medium-size MHS problems.
  - FMEA samples: real-world data, shallow but wider graph structure, usually involing a few (but) large-size MHS problems.
- We implement our method as two versions: **Dense matrix** and **Sparse matrix** in Python 3.7 programming language (using Numpy and Scipy for matrices representation and computation). For large-size MHS problems, which have more than 50,000 posible combinations, we use MHS enumerator provided by PySAT [8].
- Other competitors are *ATMS*, *ASP*, *CF*, *HS-DAG* and $HS\text{-}DAG_{QX}$

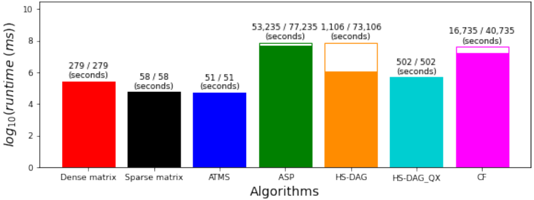
[6] Koitz-Hristov and Wotawa, "Applying algorithm selection to abductive diagnostic reasoning", 2018

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

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

Experimental Results



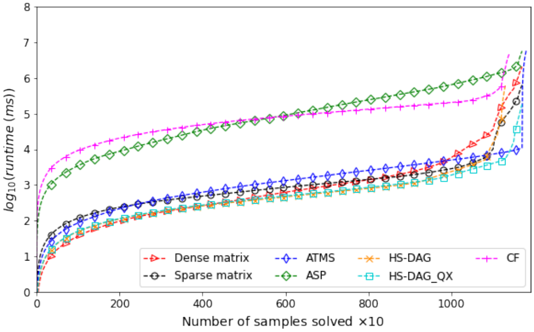


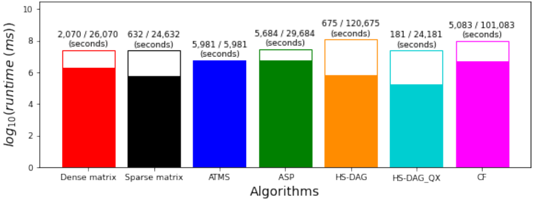
Artificial samples I benchmark dataset (166 files)

	mean	std	min	max
$\lvert\mathbb{H}\rvert$	275.07	167.12	10.00	504.00
$\lvert\mathcal{L}' \setminus \mathbb{H}\rvert$	1903.23	1504.90	6.00	6466.00
$\lvert\mathbb{T}'\rvert$	2951.10	2131.57	11.00	7187.00
$\lvert\mathbb{O}\rvert$	2.86	1.38	1.00	5.00
$\lvert P\rvert$	2088.32	1584.48	11.00	6601.00
$\lvert P_{And}\rvert$	1188.63	1349.59	8.00	6375.00
$\lvert P_{Or}\rvert$	899.69	839.58	3.00	3345.00
$\lvert\mathbb{P}\rvert$	2372.36	1730.91	24.00	7148.00
$\eta_z({M_P}^T)$	6354.90	4902.87	50.00	22307.00
$sparsity({M_P}^T)$	0.99	0.02	0.90	1.00
$max(\lvert M\rvert)$	250.34	1729.52	1.00	16866.00
$max(\eta_z(M))$	5138.28	37776.87	1.00	428754.00
$min(sparsity(M))$	0.98	0.05	0.68	1.00
max_iter	4.63	5.36	2.00	65.00
$\lvert\mathbb{E}\rvert$	2.77	5.06	1.00	50.00

(*) Actual solving time vs Penalized time (40 minutes for each unresolved run)

Experimental Results



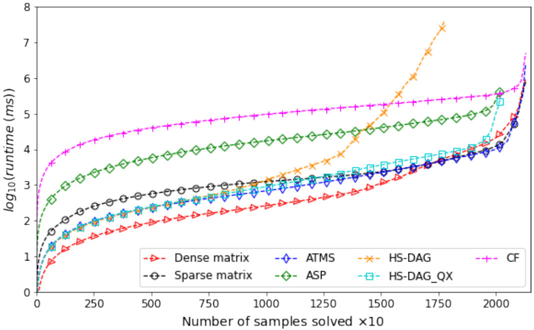


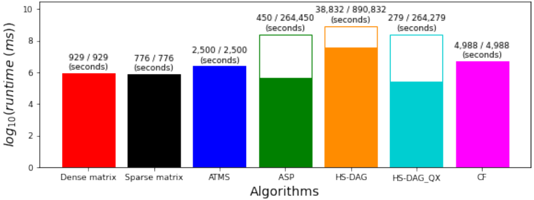
Artificial samples II benchmark dataset (118 files)

	mean	std	min	max
$\lvert\mathbb{H}\rvert$	120.42	74.35	12.00	235.00
$\lvert\mathcal{L}' \setminus \mathbb{H}\rvert$	252.74	220.50	13.00	1055.00
$\lvert\mathbb{T}'\rvert$	417.70	320.56	21.00	1147.00
$\lvert\mathbb{O}\rvert$	2.72	1.71	1.00	13.00
$\lvert P\rvert$	321.86	252.64	18.00	1110.00
$\lvert P_{And}\rvert$	201.86	186.64	9.00	1007.00
$\lvert P_{Or}\rvert$	119.99	107.40	4.00	437.00
$\lvert\mathbb{P}\rvert$	450.89	318.33	38.00	1397.00
$\eta_z(M_P{}^T)$	1180.36	861.83	83.00	4117.00
$sparsity(M_P{}^T)$	0.99	0.01	0.90	1.00
$max(\lvert M\rvert)$	16494.04	149787.13	1.00	1618050.00
$max(\eta_z(M))$	390900.36	3240888.43	1.00	34882765.00
$min(sparsity(M))$	0.94	0.08	0.59	1.00
max_iter	6.56	8.56	2.00	58.00
$\lvert\mathbb{E}\rvert$	499.60	5386.87	1.00	58520.00

(*) Actual solving time vs Penalized time (40 minutes for each unresolved run)

Experimental Results





FMEA benchmark dataset (213 files)

	mean	std	min	max
$\lvert\mathbb{H}\rvert$	26.16	20.81	3.00	90.00
$\lvert\mathcal{L}' \setminus \mathbb{H}\rvert$	27.58	19.32	6.00	84.00
$\lvert\mathbb{T}'\rvert$	71.59	75.88	13.00	299.00
$\lvert\mathbb{O}\rvert$	10.79	6.94	1.00	29.00
$\lvert P\rvert$	27.58	19.32	6.00	84.00
$\lvert P_{And}\rvert$	16.10	9.23	1.00	43.00
$\lvert P_{Or}\rvert$	11.48	11.01	1.00	41.00
$\lvert\mathbb{P}\rvert$	53.74	39.59	9.00	174.00
$\eta_z({M_P}^T)$	107.54	98.57	18.00	413.00
$sparsity({M_P}^T)$	0.95	0.04	0.73	0.99
$max(\lvert M\rvert)$	2126.49	15512.54	1.00	154440.00
$max(\eta_z(M))$	43738.87	334393.40	1.00	3459456.00
$min(sparsity(M))$	0.79	0.13	0.46	0.99
max_iter	1.94	0.24	1.00	2.00
$\lvert\mathbb{E}\rvert$	68.89	272.54	1.00	2288.00

(*) Actual solving time vs Penalized time (40 minutes for each unresolved run)

Experimental Results

Discussions about the experimental results:

- The linear algebraic approaches take advantages on problems that have wider and not too deep graph structure.
- *Sparse matrix* is the most stable algorithm in terms of **std** among those having the highest **#solved**.
- The sparsity level of interpretation matrix also has significant impact on performance.

Outline

Conclusions and Future Works

We have proposed a linear algebraic approach for solving PHCAP.

- Defining the abductive matrix to realize abductive reasoning.
- The proposed method is competitive with other existing methods.
- There are many rooms for further improvement using a more powerful BLAS.

Some related works

- Aspis, Broda, and Russo, "Tensor-Based Abduction in Horn Propositional Programs", 2018
- Sato, Inoue, and Sakama, "Abducing Relations in Continuous Spaces.", 2018

Conclusions and Future Works

Future works

- Working on a parallel implementation of the algorithm.
- Taking the MHS problem into account in vector space.
- We can consider a more compiled method for handling consistency.
- Further extend to deal with probabilistic logic.

References I

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

Aspis, Yaniv, Krysia Broda, and Alessandra Russo. "Tensor-Based Abduction in Horn Propositional Programs". In: *ILP 2018*. Vol. 2206. CEUR Workshop Proceedings. 2018, pp. 68–75.

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.

References II

- 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.
- 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.
- Sato, Taisuke, Katsumi Inoue, and Chiaki Sakama. "Abducing Relations in Continuous Spaces.". In: *IJCAI*. 2018, pp. 1956–1962.
- Selman, Bart and Hector J Levesque. "Abductive and Default Reasoning: A Computational Core". In: *AAAI*. 1990, pp. 343–348.

Thank you for your attention