Linear Algebraic Characterization of Logic Programs

Chiaki Sakama (Wakayama Univ., Japan)

Katsumi Inoue (NII, Japan)

Taisuke Sato (AIST, Japan)

KSEM 2017, Melbourne

Why LP in LA?

- Linear algebra is at the core of many applications of scientific computation, and **integrating linear algebraic and symbolic computation** is a challenging topic in AI.
- Linear algebraic computation has potential to cope with **Web scale symbolic data**, and several studies develop scalable techniques to process **huge relational knowledge bases**.
- The next challenge is applying LA to relational facts with **rules**, which would enable us to use **efficient (parallel) algorithms** of numerical linear algebra for computing LP.

Logical Reasoning in LA

- Grefenstette (2013) introduces tensor-based predicate calculus.
- Yang, *et al*. (2015) mine Horn clauses from relational facts in a vector space.
- Serafini, *et al*. (2016) integrate deductive reasoning and relational learning in logic tensor networks.
- Sato (2017) formalizes FOL in vector spaces and realizes efficient computation of Datalog.

! These studies do not target at computing LP semantics.

Contribution

1. A propositional Herbrand base is represented in a **vector space** and if-then rules in a logic program are encoded in a **matrix**.
2. The least model of a (propositional) Horn logic program is computed using **matrix products**.
3. Disjunctive logic programs are represented in **3rd-order tensors** and their minimal models are computed by algebraic manipulation of tensors.
4. Normal logic programs are represented by **3rd-order tensors** in terms of disjunctive LPs, and stable models are computed using **tensor products**.

Tensor Logic Programming

- A **Horn program** is a finite set of **rules** of the form

  $$h \leftarrow b_1 \wedge \cdots \wedge b_m \qquad (\mathrm{m} \geq 0)$$

  where h and b_i are propositional variables.

- Given a rule r of the above form, $head(r)=h$ and $body(r)=\{ b_1 , \ldots, b_\mathrm{m} \}$.
- A rule $h \leftarrow \top$ is a **fact** where $\top$ represents **true**.
- A rule $\bot \leftarrow b_1 \wedge \cdots \wedge b_\mathrm{m}$ is a **constraint** where $\bot$ represents **false**.
- The set of all propositional variables appearing in a program P is the **Herbrand base** of P (written B_P).

T_{P} Operator

- Given an interpretation I s.t. $\{\top\} \subseteq I \subseteq B_{\mathrm{P}}$, a **mapping** $T_{\mathrm{P}}: 2^{B\mathrm{p}} \rightarrow 2^{B\mathrm{p}}$ is defined as
  $T_{\mathrm{P}}(I) = \{\, h \mid h \leftarrow b_1 \wedge \cdots \wedge b_{\mathrm{m}} \in P$ and $\{b_1, \ldots, b_{\mathrm{m}}\} \subseteq I \,\}$
  if $\perp \notin I$; otherwise, $T_{\mathrm{P}}(I) = B_{\mathrm{P}}$
- The **powers** of T_{P} are defined as
  $T_{\mathrm{P}}^{\mathrm{k+1}}(I) = T_{\mathrm{P}}(T_{\mathrm{P}}^{\mathrm{k}}(I))$ ($\mathrm{k} \geq 0$) and $T_{\mathrm{P}}^{0}(I) = I$
- Given $\{\top\} \subseteq I \subseteq B_{\mathrm{P}}$ there is a **fixpoint** $T_{\mathrm{P}}^{\mathrm{n+1}}(I) = T_{\mathrm{P}}^{\mathrm{n}}(I)$ ($\mathrm{n} \geq 0$).
- For a definite program, the fixpoint $T_{\mathrm{P}}^{\mathrm{n}}(\{\top\})$ coincides with the **least model** of P.

Multiple Definitions (MD) Condition

- We assume a Horn program satisfying the condition:

  for any two rules r_1 and r_2 in P, head(r_1)=head(r_2) implies |body(r_1)|≤1 and |body(r_2)|≤1

  i.e., if two different rules have the same head, those rules contain at most one atom in their bodies.

- Every Horn program *P* is transformed to a semantically equivalent program *P'* that satisfies the MD condition.

Example

- P ={ $p \leftarrow q \wedge r$, $p \leftarrow r \wedge s$, $p \leftarrow t$, $r \leftarrow t$, $s \leftarrow$, $t \leftarrow$ }
  is transformed to
  P' ={ $p1 \leftarrow q \wedge r$, $p2 \leftarrow r \wedge s$, $p \leftarrow t$, $r \leftarrow t$, $s \leftarrow$, $t \leftarrow$,
  $p \leftarrow p1$, $p \leftarrow p2$ }
  where $p1$ and $p2$ are new propositional variables.
- P' has the least model M'={ p, $p2$, r, s, t } and $M' \cap B_P$ ={ p, r, s, t } is the least model of P.
- We consider programs satisfying the MD condition without loss of generality.

Vector Representation of Interpretations

- Let $B_{\mathrm{P}} = \{ p_1 ,\dots, p_{\mathrm{n}}\}$ be the Herbrand base. An interpretation I $(\{\top\}\subseteq I\subseteq B_{\mathrm{P}})$ of a program P is represented by a column vector $\boldsymbol{v}=(a_1 ,\dots,a_{\mathrm{n}})^{\mathrm{T}} \in \mathbf{R}^{\mathrm{n}}$ where each a_i represents the truth value of the proposition p_i such that
  - $a_i = 1$ if $p_i \in I$ $(1\leq i\leq \mathrm{n})$
  - $a_i = 0$ otherwise
- The vector representing $I=\{\top\}$ is written by $\boldsymbol{v_0}$
- We write $row_i(\boldsymbol{v})= p_i$

Matrix Representation of Horn Programs

- Let P be a Horn program and $B_{\mathrm{P}} = \{\, p_1 ,\ldots, p_{\mathrm{n}} \,\}$. P is represented by a matrix $\boldsymbol{M}_{\mathrm{P}} \in \mathbf{R}^{\mathrm{n}\times\mathrm{n}}$ s.t. for each element a_{ij} $(1\leq i, j\leq \mathrm{n})$ in $\boldsymbol{M}_{\mathrm{P}}$,
  - $a_{ij} = 1$ if $p_i = \top$ or $p_j = \bot$
  - $a_{ij_k} = 1/\mathrm{m}$ $(1\leq k\leq \mathrm{m};\ 1\leq i, j_k\leq \mathrm{n})$ if $p_{\mathrm{i}} \leftarrow p_{\mathrm{j1}} \wedge \cdots \wedge p_{\mathrm{jm}}$ is in P
  - otherwise $a_{ij} = 0$
- We write $row_i(\boldsymbol{M}_{\mathrm{P}}) = p_i$ and $col_j(\boldsymbol{M}_{\mathrm{P}}) = p_j$

Example

- $P = \{\, p \leftarrow q,\quad p \leftarrow r,\quad q \leftarrow r \wedge s,\quad r \leftarrow \top,\quad \bot \leftarrow q \,\}$ with $B_{\mathrm{P}} = \{\, p, q, r, s, \top, \bot \,\}$ is represented by $\boldsymbol{M}_{\mathrm{P}} \in \mathbf{R}^{6 \times 6}$:

$$\begin{array}{cccccccc}
 & & & & \textbf{body} & & & \\
 & & p & q & r & s & \top & \bot \\
 & p & 0 & 1 & 1 & 0 & 0 & 1 \\
\textbf{h} & q & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 1 \\
\textbf{e} & r & 0 & 0 & 0 & 0 & 1 & 1 \\
\textbf{a} & s & 0 & 0 & 0 & 0 & 0 & 1 \\
\textbf{d} & \top & 1 & 1 & 1 & 1 & 1 & 1 \\
 & \bot & 0 & 1 & 0 & 0 & 0 & 1
\end{array}$$

$p \leftarrow q,\quad p \leftarrow r,\quad p \leftarrow \bot (\equiv \top)$

$q \leftarrow r \wedge s,\quad q \leftarrow \bot (\equiv \top)$

$r \leftarrow \top (\equiv r \leftarrow),\quad r \leftarrow \bot (\equiv \top)$

$s \leftarrow \bot (\equiv \top)$

$\top \leftarrow p (\equiv \top),\ \top \leftarrow q,\ \ldots,\ \top \leftarrow \bot$

$\bot \leftarrow q (\equiv \leftarrow q),\ \bot \leftarrow \bot (\equiv \top)$

- $row_1(\boldsymbol{M}_{\mathrm{P}}) = p$ and $col_2(\boldsymbol{M}_{\mathrm{P}}) = q$

Need of MD-condition

- $P_1=\{\, p\leftarrow q\wedge r,\quad p\leftarrow s\wedge t \,\}$ is represented by

$$\begin{array}{c|ccccc} & p & q & r & s & t \\ \hline p & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ q & 0 & 0 & 0 & 0 & 0 \\ r & 0 & 0 & 0 & 0 & 0 \\ s & 0 & 0 & 0 & 0 & 0 \\ t & 0 & 0 & 0 & 0 & 0 \end{array}$$

- The matrix representation does not distinguish P_1, $P_2=\{\, p\leftarrow q\wedge s,\quad p\leftarrow r\wedge t \,\}$ and $P_3=\{\, p\leftarrow q\wedge t,\quad p\leftarrow r\wedge s \,\}$.
- Then P_1 is transformed to $P_1'=\{\, p1\leftarrow q\wedge r,\quad p2\leftarrow s\wedge t,\ p\leftarrow p1,\ p\leftarrow p2 \,\}$ which is represented by

$$\begin{array}{c|ccccccc} & p & p1 & p2 & q & r & s & t \\ \hline p & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ p1 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ p2 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ q & & & & & & & \\ r & & & & \mathbf{0} & & & \\ s & & & & & & & \\ t & & & & & & & \end{array}$$

Product

- Given a matrix $\boldsymbol{M}_{\mathrm{P}} \in \mathbf{R}^{\mathrm{n} \times \mathrm{n}}$ representing a program and a vector $\boldsymbol{v} \in \mathbf{R}^{\mathrm{n}}$ representing an interpretation $I \subseteq B_{\mathrm{P}}$, the product $\boldsymbol{M}_{\mathrm{P}} \bullet \boldsymbol{v} = (a_1, \ldots, a_{\mathrm{n}})^{\mathrm{T}}$ is computed.
- Transform $\boldsymbol{M}_{\mathrm{P}} \bullet \boldsymbol{v}$ to a vector $\boldsymbol{w} = (a'_1, \ldots, a'_{\mathrm{n}})^{\mathrm{T}}$ where $a'_i = 1$ $(1 \leq i \leq \mathrm{n})$ if $a_{ij} \geq 1$; otherwise, $a'_i = 0$
- We write $\boldsymbol{w} = \boldsymbol{M}_{\mathrm{P}} \underline{\bullet} \boldsymbol{v}$

Example (cont.)

- $P=\{\, p\leftarrow q,\quad p\leftarrow r,\quad q\leftarrow r\wedge s,\quad r\leftarrow,\quad \leftarrow q \,\}$
- Given $\boldsymbol{v}=(0,1,1,0,1,0)^{\mathrm{T}}$ representing $I=\{q, r, \top\}$,

$$\boldsymbol{M}_{\mathrm{P}} \bullet \boldsymbol{v} = \begin{array}{c} \begin{array}{cccccc} \boldsymbol{p} & \boldsymbol{q} & \boldsymbol{r} & \boldsymbol{s} & \top & \bot \end{array} \\ \begin{pmatrix} \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \frac{\mathbf{1}}{\mathbf{2}} & \frac{\mathbf{1}}{\mathbf{2}} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix} \end{array} \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{2} \\ \frac{\mathbf{1}}{\mathbf{2}} \\ \mathbf{1} \\ \mathbf{0} \\ \mathbf{3} \\ \mathbf{1} \end{pmatrix} \begin{array}{c} \boldsymbol{p} \\ \boldsymbol{q} \\ \boldsymbol{r} \\ \boldsymbol{s} \\ \top \\ \bot \end{array}$$

- Then $\boldsymbol{w} = \boldsymbol{M}_{\mathrm{P}}\, \underline{\bullet}\, \boldsymbol{v} =(1,0,1,0,1,1)^{\mathrm{T}}$ represents $J=\{p, r, \top, \bot\}$

Deduction by Matrix Product

- **<u>Proposition</u>** Let P be a Horn program and $\boldsymbol{M}_{\mathrm{P}} \in \mathbf{R}^{n \times n}$ its matrix representation. Let $\boldsymbol{v} \in \mathbf{R}^{n}$ be a vector representing $I \subseteq B_{\mathrm{P}}$. Then $\boldsymbol{w} \in \mathbf{R}^{n}$ is a vector representing $J = T_{\mathrm{P}}(I)$ iff $\boldsymbol{w} = \boldsymbol{M}_{\mathrm{P}} \, \underline{\bullet} \boldsymbol{v}$

Fixpoint Computation

- Given a matrix $\boldsymbol{M}_{\mathrm{P}} \in \mathbf{R}^{n \times n}$ and a vector $\boldsymbol{v} \in \mathbf{R}^{n}$, define

$$\boldsymbol{M}_{\mathrm{P}}\ \underline{\bullet}^{k+1}\, \boldsymbol{v} = \boldsymbol{M}_{\mathrm{P}}\ \underline{\bullet}(\boldsymbol{M}_{\mathrm{P}}\ \underline{\bullet}^{k}\, \boldsymbol{v}) \quad \text{and} \quad \boldsymbol{M}_{\mathrm{P}}\ \underline{\bullet}^{1}\, \boldsymbol{v} = \boldsymbol{M}_{\mathrm{P}}\ \underline{\bullet}\ \boldsymbol{v}\ (k \geq 1)$$

- When $\boldsymbol{M}_{\mathrm{P}}\ \underline{\bullet}^{k+1}\, \boldsymbol{v} = \boldsymbol{M}_{\mathrm{P}}\ \underline{\bullet}^{k}\, \boldsymbol{v}$ for some $k \geq 1$, write $\mathbf{FP}(\boldsymbol{M}_{\mathrm{P}}\ \underline{\bullet}\ \boldsymbol{v}) = \boldsymbol{M}_{\mathrm{P}}\ \underline{\bullet}^{k}\, \boldsymbol{v}$

Computing Least Model by Matrix Product

- **Theorem** Let P be a Horn program and $\boldsymbol{M}_{\mathrm{P}} \in \mathbf{R}^{\mathrm{n}\times\mathrm{n}}$ its matrix representation. Then $\boldsymbol{m} \in \mathbf{R}^{\mathrm{n}}$ is a vector representing the least model of P iff $\boldsymbol{m} = \mathbf{FP}(\boldsymbol{M}_{\mathrm{P}} \,\underline{\bullet}\, \boldsymbol{v_0})$ and $a_i = 1$ implies $row_i(\boldsymbol{m}) \neq \perp$ for any element a_i in $\boldsymbol{m}$

- **Corollary** P is inconsistent iff a vector $\boldsymbol{w} = \boldsymbol{M}_{\mathrm{P}}^{k} \,\underline{\bullet}\, \boldsymbol{v_0}$ ($k \geq 1$) has an element $a_i = 1$ ($1 \leq i \leq \mathrm{n}$) such that $row_i(\boldsymbol{w}) = \perp$

Example (cont.)

- $P=\{\ p\leftarrow q,\quad p\leftarrow r,\quad q\leftarrow r\wedge s,\quad r\leftarrow,\quad \leftarrow q\ \}$

$$M_P\bullet v_0 = \begin{array}{c|cccccc} & p & q & r & s & \top & \bot \\ \hline p & 0 & 1 & 1 & 0 & 0 & 1 \\ q & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 1 \\ r & 0 & 0 & 0 & 0 & 1 & 1 \\ s & 0 & 0 & 0 & 0 & 0 & 1 \\ \top & 1 & 1 & 1 & 1 & 1 & 1 \\ \bot & 0 & 1 & 0 & 0 & 0 & 1 \end{array} \begin{pmatrix}0\\0\\0\\0\\1\\0\end{pmatrix} = \begin{pmatrix}0\\0\\1\\0\\1\\0\end{pmatrix} = M_P\,\underline{\bullet}^1\, v_0$$

$$M_P\bullet(M_P\,\underline{\bullet}^1\, v_0) = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix}0\\0\\1\\0\\1\\0\end{pmatrix} = \begin{pmatrix}1\\ \frac{1}{2}\\1\\0\\2\\0\end{pmatrix} \qquad M_P\,\underline{\bullet}^2\, v_0 = \begin{pmatrix}1\\0\\1\\0\\1\\0\end{pmatrix}$$

$$= \mathbf{FP}(M_P\,\underline{\bullet}\ v_0)$$

- $\mathbf{FP}(M_P\,\underline{\bullet}\ v_0)=(1,0,1,0,1,0)^{\mathsf{T}}$ represents the least model $\{p,\ r, \top\}$ of P.

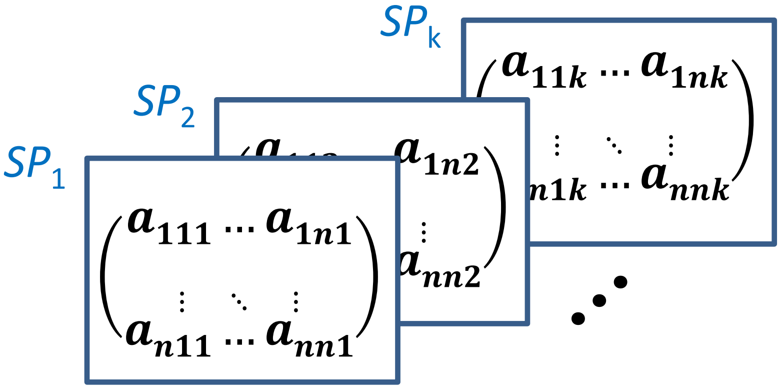
Computing Disjunctive Logic Programs by 3rd-Order Tensor

1. Split a disjunctive program into Horn programs.

   ex. $P = \{\ p \vee r \leftarrow s,\ \ q \vee r \leftarrow\ \}$ is split into

   $SP_1 = \{\ p \leftarrow s,\ \ q \leftarrow\ \}$, $SP_2 = \{\ p \leftarrow s,\ \ r \leftarrow\ \}$,
   $SP_3 = \{\ r \leftarrow s,\ \ q \leftarrow\ \}$, $SP_4 = \{\ r \leftarrow s,\ \ r \leftarrow\ \}$.

2. Represent split programs by a 3rd-order tensor.



SP_k
SP_2
SP_1
$\begin{pmatrix} a_{11k} & \cdots & a_{1nk} \\ \vdots & \ddots & \vdots \\ a_{n1k} & \cdots & a_{nnk} \end{pmatrix}$
a_{1n2}
a_{nn2}
$\begin{pmatrix} a_{111} & \cdots & a_{1n1} \\ \vdots & \ddots & \vdots \\ a_{n11} & \cdots & a_{nn1} \end{pmatrix}$

3. Compute least models of split programs by tensor product and select minimal models among them.

Computing Normal Logic Programs by 3rd-Order Tensor

- Transform a normal program to a semantically equivalent disjunctive program (Fernandez, *et al.*, 1993).

$$h \leftarrow b_1 \wedge \cdots \wedge b_m \wedge \neg b_{m+1} \wedge \cdots \wedge \neg b_n$$

$$\Downarrow$$

$$h \vee \varepsilon b_{m+1} \vee \cdots \vee \varepsilon b_n \leftarrow b_1 \wedge \cdots \wedge b_m$$

and $\varepsilon p \leftarrow p$ for $p \in B_P \backslash \{\top, \bot\}$

+ integrity constraints $\varepsilon p \Rightarrow p$ for $p \in B_P \backslash \{\top, \bot\}$

where εp is a new atom associated with p.

- Compute stable models via minimal models of the transformed disjunctive program.

Complexity

- The least model of a Horn program is computed in O(N) time and space where N is the size of the program (Dowling& Gallier, 1984).
- The proposed method requires O(n^2) space and O(n^4) time in the worst case where n is the number of propositional variables in B_{P}
- Since the size of a matrix is independent of the size of a program, LA computation would be advantageous in a large knowledge base on a fixed language.

Conclusion

- Linear algebraic characterization of logic programs bridges symbolic and linear algebraic approaches, which would contribute to a step for realizing logical inference in huge scale of knowledge bases.
- We are now implementing/evaluating the algorithm and also plan to use parallel computing on GPU.