Learning Disjunctive Logic Programs from Interpretation Transition

Yi Huang[1,3] Yisong Wang[1] Ying Zhang[2]
and Mingyi Zhang[2]

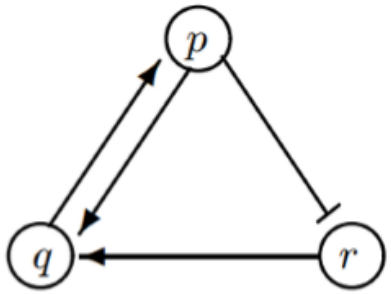
[1]Guizhou University
[2]Guizhou Academy of sciences
[3]Chongqing University of Arts and Sciences

September 4, ILP2016

Dynamics is involved in [wikipedia]:

- Mathematics and Computer Science:
  - A **dynamical system** is a system in which a function describes the time dependence of a point in a space.
  - At any given time a dynamical system has a state that can be represented by a point in an appropriate state space.
  - The *evolution rule* of the dynamical system is a function that describes what future states follow from the current state.

Boolean networks is an important tool to model dynamical systems!



$$p^{t+1} = q^t,$$
$$q^{t+1} = p^t \wedge r^t,$$
$$r^{t+1} = \neg p^t.$$

Figure: A Boolean network N

- Inoue proposed Synchronous Boolean Network can be direct translated into a normal logic program(NLP).
  - discarding its time parameter t, and
  - replacing "=" by "$\leftarrow$".

- Inoue and Ribeiro and Sakama proposed a novel framework to learn NLP from transitions of interpretations.
  - naive/ground resolution
  - LF1T algorithm

    Input: observations/examples $E \subseteq 2^{\mathcal{A}} \times 2^{\mathcal{A}}$, and a background program B,
    Output: A NLP P such that $T_{P \cup B}(I) = J$ for each $\langle I, J \rangle \in E$.
    $P \leftarrow \emptyset$;
    for each $A \in J$,

    $$R_I^A := (A \leftarrow \bigwedge (I \cup \neg \bar{I})).$$

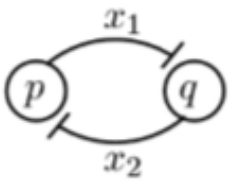
    Adding R_A^I into P.
    Simplifying P using B by ground resolution and subsumption.

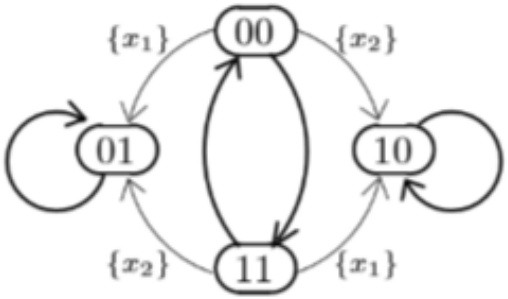
Questions:

A (asychronous) Boolean network (ABN) differs from SBNs only in the state update:

Not all nodes are necessary updated at the same time!
The next state of the current state is nondeterministic!

$$p^{t+1} = \neg q^t,$$
$$q^{t+1} = \neg p^t.$$





We consider that the next states are all minimal in an ABN!
That can be captured by a disjunctive logic program!

- An observation/example is a state transition $\langle I, J\rangle \in 2^{\mathcal{A}} \times 2^{\mathcal{A}}$,
- A set of observations/examples E is *coherent* if J and J' are incomparable for every two pairs $\langle I, J\rangle$ and $\langle I, J'\rangle$ of E.
- E is *consistent* with a logic program B if, for each $\langle I, J\rangle \in E$, $I \models bd(r)$ implies $J \models hd(r)$ for every $r \in B$.

Learning from nondeterministic 1-step Transition (LFDT) task:

Input: coherent observations/examples $E \subseteq 2^{\mathcal{A}} \times 2^{\mathcal{A}}$, which is consistent with the background program B,

Output: A DLP P such that $J \in T^d_{P\cup B}(I)$ for each $\langle I, J\rangle \in E$.

Algorithm 1: LFDT(E, B)

input : A set E of state transitions and a background theory B such that E is coherent
output: A logic program P

$P \leftarrow B$
foreach $\langle I, J\rangle \in E$ **do**
  $Q \leftarrow \{R_I^q | q \in J\}$
  $E \leftarrow E \setminus \{\langle I, J\rangle\}$
  foreach $\langle I', J'\rangle \in E$ *with* $I' = I$ **do**
    $E \leftarrow E \setminus \{\langle I', J'\rangle\}$
    foreach $p \in J'$ *and* $r \in Q$ **do**
      $Q \leftarrow Q \cup \{hd(r) \cup \{p\} \leftarrow bd(r)\}$
    end
  end
  foreach $r \in Q$ **do** $AddRule(r, P)$
end
$P \leftarrow P \setminus B$
return P

Algorithm 2: AddRule(r, P)

Input: A rule r and a logic program P
if $\exists r' \in P$ *s.t.* $r' \prec r$ **then return**;
foreach $r' \in P$ **do** **if** $r \prec r'$ **then** $P \leftarrow P \setminus \{r'\}$;
$P \leftarrow P \cup \{r\}$;
while $r, r_1, \ldots, r_k \in P$ *are combined resolvable* **do**
  AddRule($cr(r, r_1, \ldots, r_k), P$);
end
foreach $r' \in P$ **do**
  if r *is disjunctively ground resolvable w.r.t.* r' **then**
    AddRule($gr(r, r'), P$);
  else if r' *is disjunctively ground resolvable w.r.t.* r **then**
    AddRule($gr(r', r), P$);
  end
end
return P;

r subsumes r', written $r \prec r'$, if $hd(r)\theta \subseteq hd(r')$ and $bd(r)\theta \subseteq bd(r')$ for some substitution θ.

Definition (disjunctive ground resolution)

Let r and r' be two rules such that

(a) $l \in bd(r)$ and $\bar{l} \in bd(r')$,

(b) $bd(r') \setminus \{\bar{l}\} \subseteq bd(r) \setminus \{l\}$, and

(c) $hd(r') \subseteq hd(r)$.

The *(disjunctive) ground resolvent* $gr(r, r')$ of r *w.r.t.* r' on l is:

$$hd(r) \leftarrow bd(r) \setminus \{l\}.$$

Given the following two rules r_1, r_2, to compute $gr(r_1, r_2)$:

$$r_1 : \quad p \vee q \leftarrow p \wedge q \wedge \neg s,$$
$$r_2 : \quad q \leftarrow q \wedge s.$$
$$gr(r_1, r_2) = p \vee q \leftarrow p \wedge q.$$

Definition (combined resolution)

Let $r_1, \ldots, r_k$ and r be the following rules:

$$r_1 : hd(r_1) \leftarrow bd^+ \cup \neg(bd^- \cup \{b_1\}),$$
$$\vdots$$
$$r_k : hd(r_k) \leftarrow bd^+ \cup \neg(bd^- \cup \{b_k\}),$$
$$r : hd(r) \leftarrow bd^+ \cup \{b_i | 1 \leq i \leq k\} \cup \neg bd^{-\prime}, \text{ such that}$$

(a) $bd^- \cap \{b_i | 1 \leq i \leq k\} = \emptyset$, and

(b) $hd(r_i) \subseteq hd(r)$ for every i $(1 \leq i \leq k)$.

The *combined resolvent* $cr(r, r_1, \ldots, r_k)$ of $r, r_1, \ldots, r_k$ is:

$$hd(r) \leftarrow bd^+ \cup \neg(bd^- \cup bd^{-\prime}). \quad (1)$$

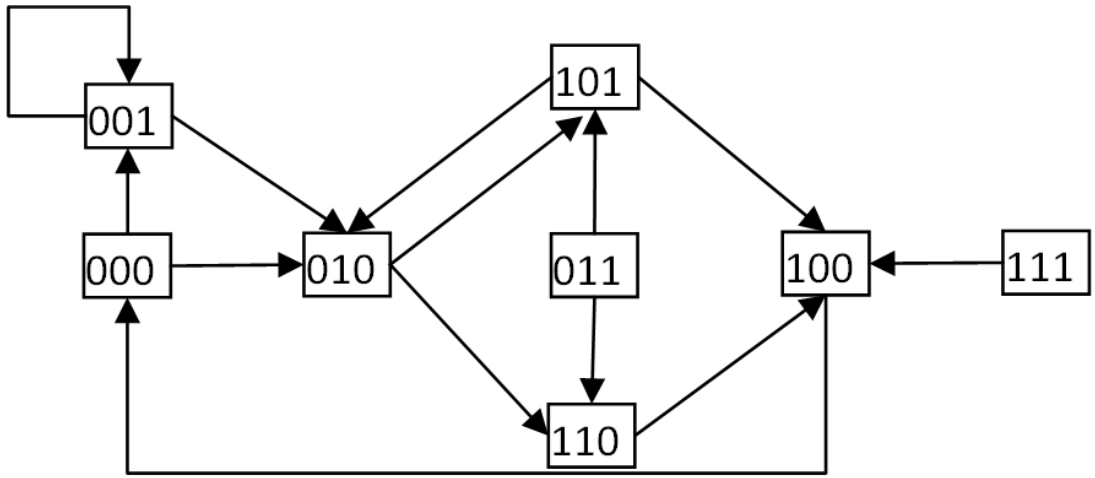


Table: Execution of **LFDT** with $B = \emptyset$ and E

step	$I \to J$	operation	rule	ID	P
1	$\epsilon \to r, \epsilon \to q$	$r^{\epsilon}_{q \vee r}$	$q \vee r \leftarrow \neg p \wedge \neg q \wedge \neg r$	1	1
2	$p \to \epsilon$	r^{p}_{ϵ}			1
3	$q \to pr, q \to pq$	r^{q}_{p}	$p \leftarrow \neg p \wedge q \wedge \neg r$	2	1,2
		$r^{q}_{p \vee r}$	$p \vee r \leftarrow \neg p \wedge q \wedge \neg r$	3	1,2
		$r^{q}_{p \vee q}$	$p \vee q \leftarrow \neg p \wedge q \wedge \neg r$	4	1,2
		$r^{q}_{q \vee r}$	$q \vee r \leftarrow \neg p \wedge q \wedge \neg r$	5	1,2,5
		$gr(1,5)$	$q \vee r \leftarrow \neg p \wedge \neg r$	6	2,6

We get the result:

$$P = \left\{ \begin{array}{l} q \vee r \leftarrow \neg p, \\ p \vee q \leftarrow p \wedge \neg q \wedge r, \\ p \leftarrow q. \end{array} \right\} \quad (2)$$

A logic program P is *complete* for E *w.r.t.* B if $\{J \mid \langle I, J\rangle \in E\} \subseteq T^d_{B\cup P}(I)$ for any $I \in E^i$, it is *sound* for E if $T^d_{B\cup P}(I) \subseteq \{J \mid \langle I, J\rangle \in E\}$ for any $I \in E^i$.

Theorem

*The algorithm **LFDT** is sound and complete (with disjunctive ground resolution, combined resolution, and/or subsumption reduction).*

What we did:

- the **LFDT** inductive learning framework for a restricted ABN
- that is both sound and complete.

Future work:

- Implementation **LFDT**,
- Experimentation,
- refinement the **LFDT** algorithm in a way of **LF1T**.