Learning any memory-less discrete semantics for dynamical systems represented by logic programs

Tony Ribeiro1,2, Maxim Falschleuter3, Morgan Magnin1, Karatsi Inoue3
(1) Université de Nantes, CNRS, LINA, F-44000 Nantes, France
(2) LIRMM, CNRS, Université de Montpellier, 34090 Montpellier, France
(3) National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

Motivations: Learning Dynamics

- Given a set of input/output states of a black-box system, learn its internal mechanics.
- Discrete system: input/output are vectors of same size which contain discrete values.
- Dynamic system: input/output are states of the system and output is the next input.

Coal: produce an artificial system with the same behavior, i.e., a digital twin.
- Representation: propositional logic programs encoding multi-valued discrete variables.
- Method: learn the dynamics of systems from its state transitions.

Formalization: MVL and DMVL

Definition 1 (Atoms): Let \( V = \{v_1, \ldots, v_n\} \) be a finite set of \( n \) variables, and \( \mathcal{D} : V \rightarrow 2^\mathcal{D} \) the domain of \( V \). An atom of \( \mathcal{D} \) is defined as \( \{v \} \) for \( v \in V \).

Definition 2 (Multi-valued logic program): A MVL is a set of \( \mathcal{D} \) rules.

Definition 3 (Dynamic, MVL): Let \( T \subseteq V \) and \( F \subseteq \mathcal{D} \) such that \( F = V \setminus T \). A MVL over \( T \) is a MVL such that \( \forall R \in P, \exists \mathbf{a} \in \mathcal{D} \) for all \( \mathbf{a} \) in \( \mathcal{D} \).

Definition 4 (Discrete state): A discrete state \( s \) on \( T \) (resp. \( F \) ) is a MVL if it is a function from \( T \) (resp. \( F \) ) to \( \mathcal{D} \).

Definition 5 (Iteration): A transition is a couple of states \((s, s') \in S \times S\).

Definition 6 (Semantics): A semantics is a function \( \text{sem} : \{\text{MVL} \} \rightarrow \{\text{SC}\} \).

Problem: Dynamics Semantics

Semantics: Decide the target states according to a DMVL and a feature state.

- Synchronous
- Asynchronous
- General

A semantics that produces the same states, when being given the atoms of its own decision is pseudo-idempotent and is compatible with its transition optimal DMVL.

Definition 8 (Pseudo-idempotent Semantics): Let \( DS \) be a dynamical semantics. \( DS \) is pseudo-idempotent if, for all \( P \) a DMVL, \( DS(P) \cap DS(P') = DS(P) \).

Algorithm: GULA

Definition 9 (Rule least specialization): Let \( R \) be a MVL rule and \( s \in S \) such that \( \mathcal{D} \). The least specialization of \( R \) by a \( \mathcal{D} \) according to \( F \) is \( A \).

Definition 10 (Program least revision): Let \( P \) be a DMVL, \( s \in S \), \( T \subseteq S \), \( F \subseteq S \) such that \( \mathcal{D} \). Let \( P' = \{A \in P \mid R \text{ conflicts with } T\} \). The least revision of \( P \) by \( T \) according to \( A \) is \( D \). If \( \mathcal{D} \), then \( D \).

Learning From Any Semantics Using Constraints

Definition 11 (Constrained DMVL): Let \( P \) be a DMVL over \( \mathcal{D} \), \( s \subseteq S \), \( T \subseteq S \), \( F \subseteq S \) such that \( \mathcal{D} \). Let \( P \) be a DMVL such that \( \mathcal{D} \). A DMVL \( P \) is a DMVL such that \( \mathcal{D} \) and \( \mathcal{D} \). A constraint over \( T \) such that \( \mathcal{D} \) and \( \mathcal{D} \). A constraint over \( T \) such that \( \mathcal{D} \) and \( \mathcal{D} \). A constraint over \( T \) such that \( \mathcal{D} \) and \( \mathcal{D} \). A constraint over \( T \) such that \( \mathcal{D} \) and \( \mathcal{D} \).

Definition 12 (Constraint transition matching): Let \( s, s' \in S \). \( \mathcal{D} \) and \( \mathcal{D} \). \( \mathcal{D} \) and \( \mathcal{D} \). \( \mathcal{D} \) and \( \mathcal{D} \). \( \mathcal{D} \) and \( \mathcal{D} \).

Definition 13 (Suitable and optimal constraints): Let \( T \subseteq S \). \( \mathcal{D} \) and \( \mathcal{D} \). \( \mathcal{D} \) and \( \mathcal{D} \). \( \mathcal{D} \) and \( \mathcal{D} \). \( \mathcal{D} \) and \( \mathcal{D} \).

Definition 14 (Synonymous constrained semantics): The synonymous constrained semantics \( \mathcal{D} \).

Contributions

- Previous works: Synchronous deterministic transitions only [1-3].
- Novelty: Learn from any memory-less discrete dynamics semantics.
- Application: semantic choice, which has an important meaning for the one who try to model a system, can now be done a posteriori. The rules can explain local interactions and constraint are tests of semantic behaviors.
- Weakness: current complete method is too costly/sensitive to deal with real system.
- Outlook: development of heuristic approach (DVMVL, PRIDE) to tackle real data and tools (see other paper) to extract knowledge from the learned model.
- The source code is available as open source on Github. See QR-code.