Learning Multi-Valued Biological Models with Delayed Influence from Time-Series Observations

Tony Ribeiro
Morgan Magnin, Katsumi Inoue, Chiaki Sakama

Department of Informatics
The Graduate University for Advanced Studies, Japan

National Institute of Informatics, Japan

Wakayama University, Japan

ILP, August 20th, 2015
Motivations

Problem:
- Time series data alone may be not sufficient, we need models.

Goal:
- Automated modeling of systems dynamics from these data.
Learning From Interpretation Transitions (LFIT)

A framework for learning system dynamics from state transitions.

**Basic Idea:**
- Learn a logic program by observing the behavior of a system.
- This logic program represents the dynamics of the system.

**Input:** Behavior of the system

**Output:** Dynamics of the system

\[
\begin{align*}
    p(t+1) &\leftarrow q(t). \\
    q(t+1) &\leftarrow p(t) \land r(t). \\
    r(t+1) &\leftarrow \neg p(t).
\end{align*}
\]

**Representation:** Logic Program
Learning From Interpretation Transitions (LFIT)

A framework for learning system dynamics from state transitions.

**Basic Idea:**
- Learn a logic program by observing the behavior of a system.
- This logic program represents the dynamics of the system.

What happens

\[
\begin{align*}
p(t+1) & \leftarrow q(t). \\
q(t+1) & \leftarrow p(t) \land r(t). \\
r(t+1) & \leftarrow \neg p(t).
\end{align*}
\]

Why it happens

**Representation:** Logic Program
Learning From $k$-Step Transitions (LF$k$T)

Motivation

- Learn Systems with delayed influences: Markov($k$) systems.
- Extract causality from observations, why things happen.
- Predict consequences of new observations, what will happen.

What is new?

- **Automatic** computation of delays
- **Multi-valued** variables
- **Real** benchmark evaluation
Outline

1 Preliminaries

2 New contribution
   - Evaluation

3 Conclusion
We consider a logic program as a set of rules of the form

\[ p \leftarrow p_1 \land \cdots \land p_m \land \neg p_{m+1} \land \cdots \land \neg p_n \]  \hspace{1cm} (1)

where \( p \) and \( p_i \)'s are atoms (\( n \geq m \geq 1 \)).

**Definition (Herbrand Base)**

The Herbrand Base of a program \( P \), denoted by \( \mathcal{B} \), is the set of all atoms in the language of \( P \).

**Example**

- \( R_1 = a \leftarrow b \land c \)
- \( R_2 = b \leftarrow a \land c \)
- \( P = \{ R_1, R_2 \} \) is a logic program
- The Herbrand Base of \( P \) is \( \mathcal{B} = \{ a, b, c \} \)
Formalization: Markov($k$) into Logic Program

Definition (Timed Herbrand base)

Let $\mathcal{B}$ be the Herbrand base of a program $P$ and $k$ be a natural number. The timed Herbrand base of $P$ (with period $k$) denoted by $\mathcal{B}_k$, is as follows:

$$\mathcal{B}_k = \bigcup_{i=1}^{k} \{v_{t-i} | v \in \mathcal{B}\}$$

Where $t$ is a constant term which represents the current time step.

Example

If the Herbrand base of a program $P$ is $\mathcal{B} = \{a, b, c\}$ then

- $\mathcal{B}_1 = \{a_{t-1}, b_{t-1}, c_{t-1}\}$
- $\mathcal{B}_2 = \{a_{t-1}, b_{t-1}, c_{t-1}, a_{t-2}, b_{t-2}, c_{t-2}\}$
Markov($k$) Systems

A Markov($k$) system can be interpreted as a logic program.

**Definition (Markov($k$) system)**

Let $P$ be a logic program, $\mathcal{B}$ be the Herbrand base of $P$ and $\mathcal{B}_k$ be the timed Herbrand base of $P$ with period $k$. A Markov($k$) system $S$ with respect to $P$ is a logic program where for all rules $R \in S$, $h(R) \in \mathcal{B}$ and all atoms appearing in $b(R)$ belong to $\mathcal{B}_k$.

**Example**

If the Herbrand base of a program $P$ is $\mathcal{B} = \{a, b\}$ then

- $\mathcal{B}_2 = \{a_{t-1}, b_{t-1}, a_{t-2}, b_{t-2}\}$.
- Let $R_1 = a \leftarrow b_{t-1}, b_{t-2}$ and $R_2 = b \leftarrow a_{t-2}, \neg b_{t-2}$.
- $S = \{R_1, R_2\}$ is a Markov(2) system.
Let $S$ be a Markov($k$) system as follows:

$$S = \{(a \leftarrow b_{t-1}, b_{t-2}), (b \leftarrow a_{t-2}, \neg b_{t-2})\}$$

Eight traces of executions of the system $S$
Outline

1 Preliminaries

2 New contribution
   • Evaluation

3 Conclusion
Example: Detection of the Delay

Let $S$ be a Markov($k$) system as follows:

$$S = \{(a \leftarrow b_{t-1}, b_{t-2}), (b \leftarrow a_{t-2}, \neg b_{t-2})\}$$

Eight traces of executions of the system $S$
Example: Detection of the Delay

Let $S$ be a Markov($k$) system as follows:

$$S = \{(a \leftarrow b_{t-1}, b_{t-2}), (b \leftarrow a_{t-2}, \neg b_{t-2})\}$$

Eight traces of executions of the system $S$
Example: Detection of the Delay

Let $S$ be a Markov($k$) system as follows:

$$S = \{(a \leftarrow b_{t-1}, b_{t-2}), (b \leftarrow a_{t-2}, \neg b_{t-2})\}$$

Eight traces of executions of the system $S$
Multivalued Variables

In order to represent multi-valued variables, we now restrict all atoms of a logic program to the form $\text{var}^{\text{val}}$. We consider a multi-valued logic program as a set of rules of the form

$$\text{var}^{\text{val}} \leftarrow \text{var}_1^{\text{val}_1} \land \cdots \land \text{var}_n^{\text{val}_n}$$

(2)

where $\text{var}^{\text{val}}$ and $\text{var}_i^{\text{val}_i}$ are atoms ($n \geq 0$).

$$S = \{(a^1 \leftarrow b_{t-1}^1, b_{t-2}^1), (b^1 \leftarrow a_{t-2}^1, b_{t-2}^0)\}$$
DREAM4 input data

- Case studies:
  - 5 different systems each composed of 10 genes
  - 5 different systems composed of 100 genes
  - All come from E. coli and yeast networks

- Data sets available for each system of 10 genes (resp. 100):
  - 5 (resp. 10) time series data with 21 time points
  - Steady state at wild type, i.e. 1 steady state
  - Steady state after knocking out each gene, i.e. 10 steady states (resp. 100)
  - Steady state after knocking down each gene (transcription rate at 50%), i.e. 10 steady states (resp. 100)
  - Steady states after some random multifactorial perturbations, i.e. 10 steady states
Case studies:
- 5 different systems each composed of 10 genes
- 5 different systems composed of 100 genes
- All come from E. coli and yeast networks

Data sets available for each system of 10 genes (resp. 100):
- 5 (resp. 10) time series data with 21 time points
- Steady state at wild type, i.e. 1 steady state
- Steady state after knocking out each gene, i.e. 10 steady states (resp. 100)
- Steady state after knocking down each gene (transcription rate at 50%), i.e. 10 steady states (resp. 100)
- Steady states after some random multifactorial perturbations, i.e. 10 steady states
DREAM4 Challenge

- **DATA:** 5 different time series of genes expression.
- **GOAL:**
  - Predict the directed unsigned interaction graph
  - Predict steady states from dual knockouts
Learning phase

- INPUT: 5 series of 20 transitions with different perturbations.
- Approach:
  - Learn independently each series with LFkT.
  - Evaluate rules on all series (full cross-validation).
INPUT:  
- An initial state  
- 5 different conditions of dual genes to be knockout simultaneously

GOAL:  
- Predict the point attractor

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>run time</th>
<th>raw output</th>
<th>final output</th>
<th>Mean squared error</th>
</tr>
</thead>
<tbody>
<tr>
<td>insilico_size10.1</td>
<td>28s</td>
<td>118,834</td>
<td>359</td>
<td>0.073</td>
</tr>
<tr>
<td>insilico_size10.2</td>
<td>2m5s</td>
<td>401,923</td>
<td>462</td>
<td>0.064</td>
</tr>
<tr>
<td>insilico_size10.3</td>
<td>44s</td>
<td>151,021</td>
<td>480</td>
<td>0.019</td>
</tr>
<tr>
<td>insilico_size10.4</td>
<td>22s</td>
<td>90,904</td>
<td>387</td>
<td>0.031</td>
</tr>
<tr>
<td>insilico_size10.5</td>
<td>1m04s</td>
<td>297,364</td>
<td>326</td>
<td>0.091</td>
</tr>
</tbody>
</table>

Evaluation: precision is evaluated as the mean squared error of the difference between predicted/expected values.
Outline

1. Preliminaries
2. New contribution
   • Evaluation
3. Conclusion
Conclusion

Contribution
- Extension of the LFkT algorithm
  - Detect delay dynamically
  - Handle multivalued variable
- Practical methods
  - Automatic discretization of time series (pre-processing)
  - Heuristic for model simplification (post-processing)

Current & Future works
- Improve performances to tackle big network (DREAM4 100)
- Design method to extract the real influences