NII OPEN HOUSE 2021

# 変遷し続ける世界にうなずきを与える


Katsumi Inoue

dblp computer science bibliography

researchmap

Google Scholar


## 機械学習と記号推論の融合

iNOUE Laboratory
inference & learning

http://research.nii.ac.jp/il/


大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

# Differentiable Logical Inference


**Akihiro Takemura**
(Doctoral student)
Google Scholar


## 1. Translate program *P* into a Program Matrix $D^P$

$$P \quad \begin{array}{l} \texttt{p :- not q.} \\ \texttt{q :- not p.} \end{array}$$

Program

$$D^P \quad \begin{array}{c|cccc} & p & q & \bar{p} & \bar{q} \\ \hline p & 0 & 0 & 0 & 1 \\ q & 0 & 0 & 1 & 0 \end{array}$$

Program Matrix

$$\boldsymbol{x} \quad \begin{array}{c|c} p & 0 \\ q & 0 \end{array}$$

Interpretation vector

## 2. Define Loss function w.r.t. continuous interpretation

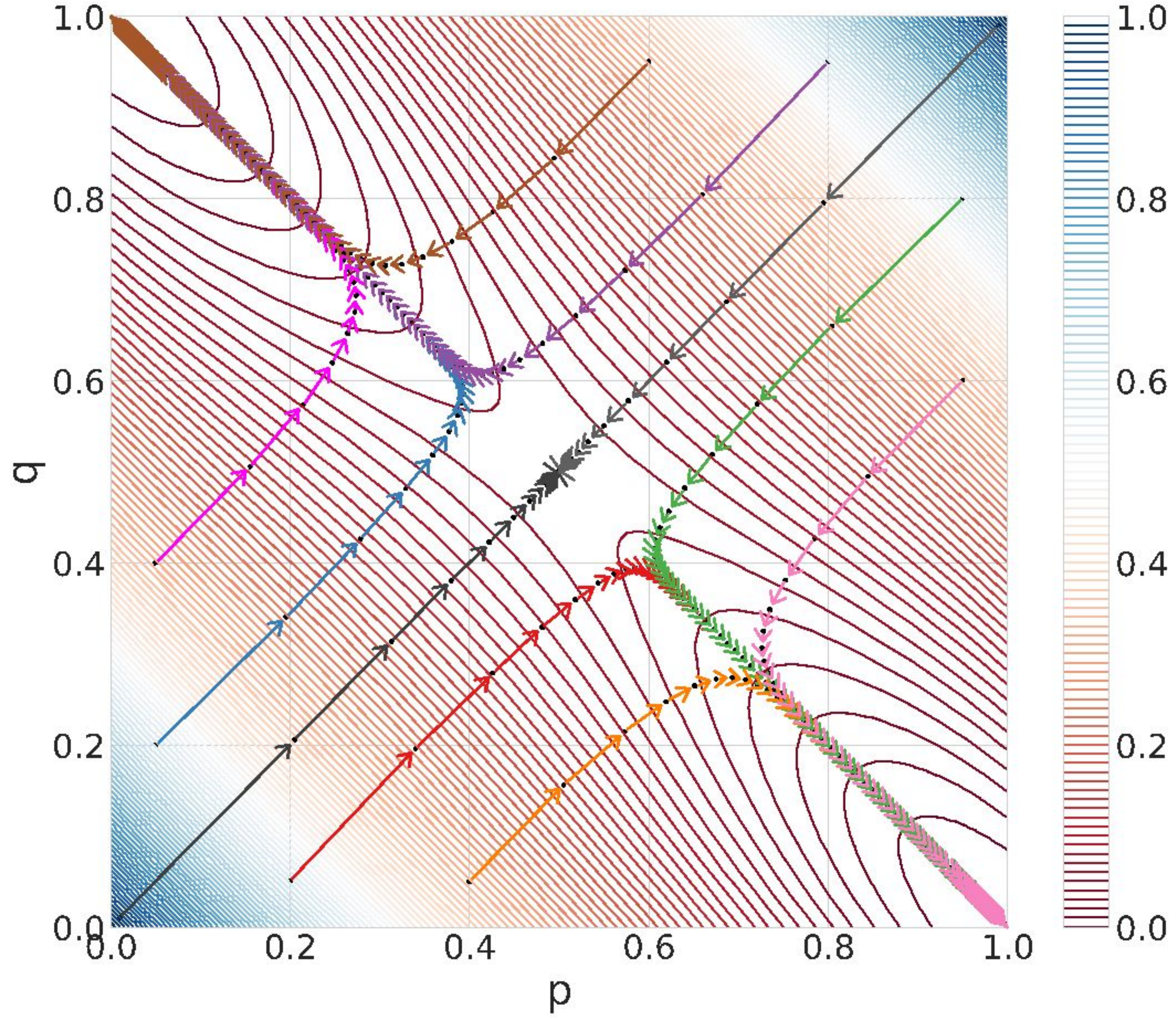
$$L(\boldsymbol{x})$$

Loss function w.r.t. ***x***

$$\frac{\partial L(\boldsymbol{x})}{\partial x}$$

Gradient of L(x) w.r.t. ***x***

## 3. Minimize the loss with gradient descent



# Rule Set Generation from Tree Ensembles with ASP

**Akihiro Takemura**
(Doctoral student)
Google Scholar

Data

$\text{class}(1) \leftarrow x_1 \leq 1 \wedge x_3 \leq 1 \wedge x_2 \leq 1 \ldots$

$\text{class}(0) \leftarrow x_2 \leq 5 \wedge x_1 \leq 3 \wedge x_4 \leq 10 \ldots$

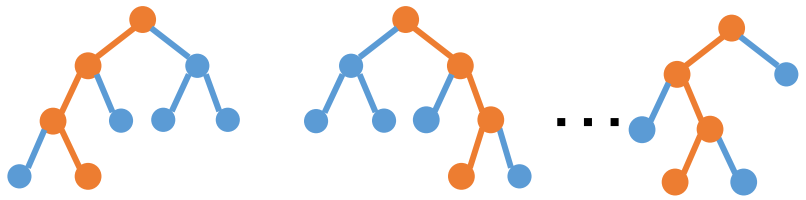
$\text{class}(2) \leftarrow x_6 \leq 3 \wedge x_4 \leq 2 \wedge x_1 \leq 1 \ldots$

...

Convert to Rules

For a tree ensemble with *K* trees, each with depth *D*, there can be (at maximum) ~$K2^{D+1}$ rules

Tree Ensemble
(ML model)



Rule Set Generation in ASP

Declarative encoding of user-defined constraints and preferences
Selecting

Important Rules

$\text{class}(1) \leftarrow$
$x_1 \leq 1 \wedge x_3 \leq 1$
$\text{class}(0) \leftarrow$
$x_6 > 1 \wedge x_5 \leq 1$
...

No. rules can be controlled
Performance metrics can be taken into consideration in encoding

# Bridging Logic and Matrix


**Taisuke Sato**

(Professor by Special Appointment)

Google Scholar  researchmap


$r_1(\mathrm{a,b}),\ r_1(\mathrm{b,c})$
$r_1(\mathrm{c,d}),\ r_1(\mathrm{d,a})$

Transitive closure program for $r_1$:

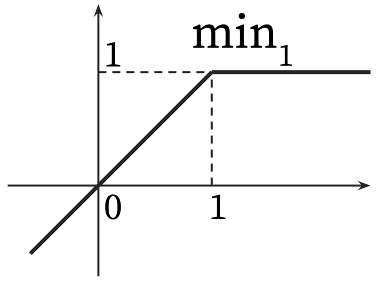
$$r_2(x,z) \leftarrow r_1(x,z)$$
$$r_2(x,z) \leftarrow r_1(x,y)\ \&\ r_2(y,z)$$

⬇

$$r_2(x,z) \Leftrightarrow r_1(x,z) \vee \exists y\,(r_1(x,y) \wedge r_2(y,z))$$

⬇

$$\mathbf{R}_2 = \min_1(\mathbf{R}_1 + \mathbf{R}_1\mathbf{R}_2)$$




$\min_1(x)=\min(x,1)$

⬇ least solution

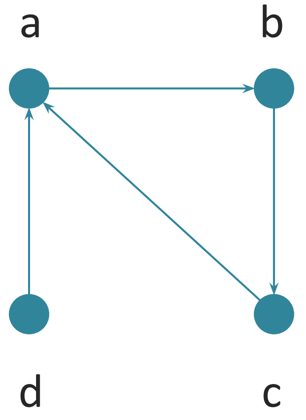
$$\tilde{\mathbf{R}}_2 = \varepsilon(\mathbf{R}_1 + \mathbf{R}_1\tilde{\mathbf{R}}_2) = (\mathbf{I} - \varepsilon\mathbf{R}_1)^{-1}\varepsilon\mathbf{R}_1.$$

⬇

$$\mathbf{R}_2 = \tilde{\mathbf{R}}_2 > 0$$

(positive entry $\rightarrow$ **1**, else **0**)

$$\mathbf{R}_1 = \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$




⬇

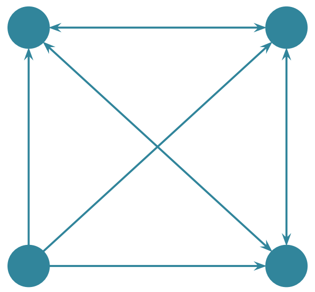
$$\|\mathbf{R}_1\|_\infty = \max\{1,1,1,1\} = 1$$
$$\epsilon = (1 + \|\mathbf{R}_1\|_\infty)^{-1} = 1/2$$

⬇

$$\begin{aligned} \tilde{\mathbf{R}}_2 &= (\mathbf{I} - \epsilon\mathbf{R}_1)^{-1}\epsilon\mathbf{R}_1 \\ &= \left( \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \\ 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 \end{pmatrix} \right)^{-1} \begin{pmatrix} 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \\ 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0.1428 & 0.5714 & 0.2857 & 0.0000 \\ 0.2857 & 0.1428 & 0.5714 & 0.0000 \\ 0.5714 & 0.2857 & 0.1428 & 0.0000 \\ 0.5714 & 0.2857 & 0.1428 & 0.0000 \end{pmatrix} \end{aligned}$$

⬇

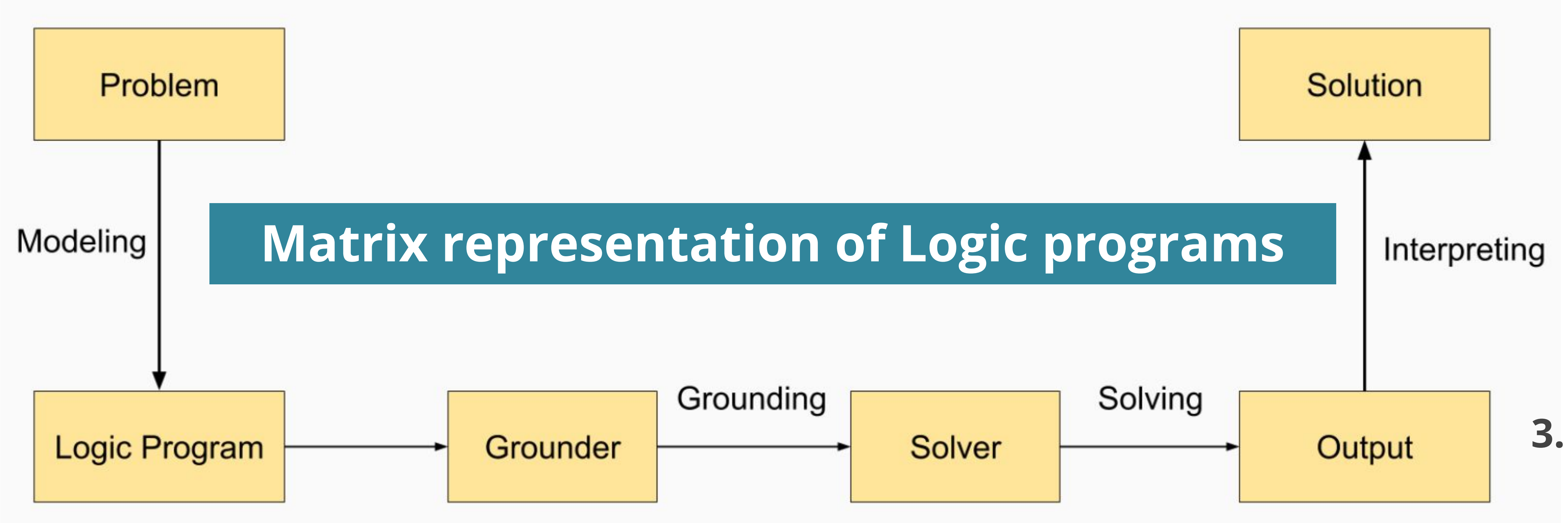
$$\mathbf{R}_2 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$



$r_2(\mathrm{d,c}) = 1$

# On Linear Algebraic Computation for Logic Programming


Tuan Nguyen Quoc
(Doctoral student)


- 否定を含む一般論理プログラムの安定モデルを線形代数手法で解く。
- 論理プログラムを行列表現。そのスパース性を活用する。
- 大規模論理プログラムをGPU代数計算により高速に解く。

$P = \{p \leftarrow q \wedge s, q \leftarrow p \wedge t, s \leftarrow \neg t, t \leftarrow, u \leftarrow v\}$

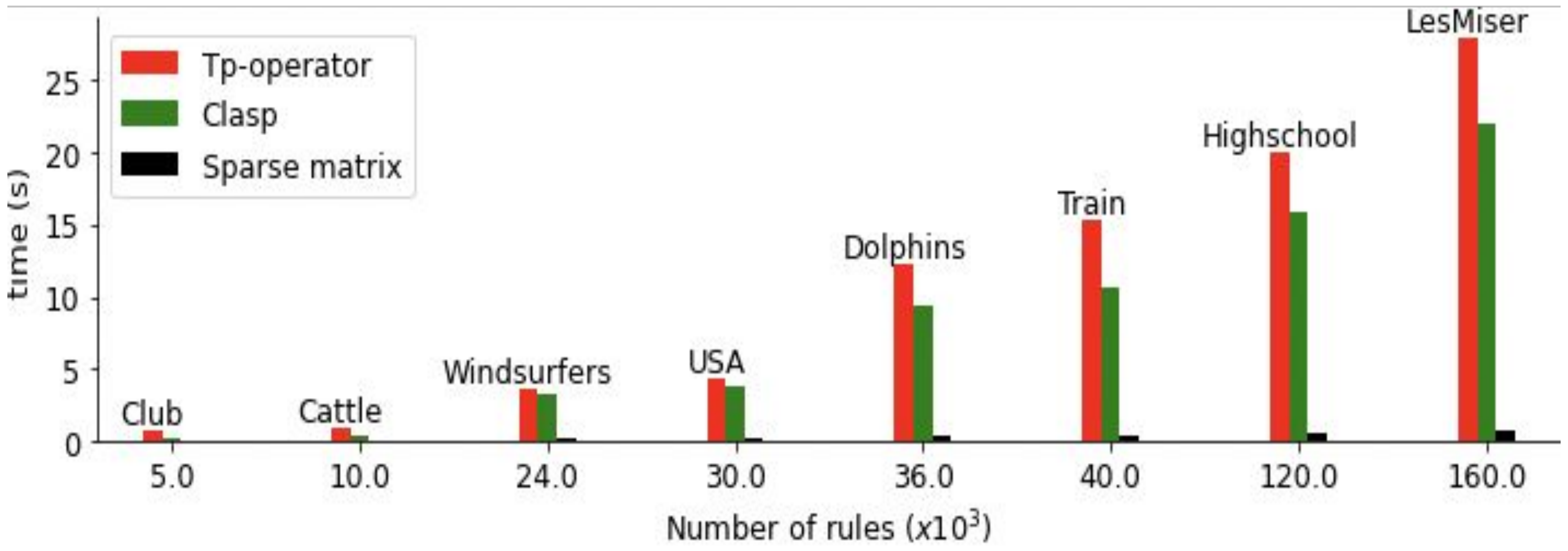
1. **Encode logic programs into matrices**
   - Scalability
   - Robustness
   - Integrability *e.g.*, ANN

$$\begin{array}{c|ccccccc} & p & q & s & t & \neg t & u & v \\ \hline p & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ q & 1/2 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ s & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ t & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \neg t & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ u & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ v & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Program matrix

2. **Perform logic reasoning in vector spaces:**
   - Deduction $\dfrac{P \rightarrow Q \quad P}{Q}$
   - Abdution $\dfrac{P \rightarrow Q \quad Q}{P}$
   - Induction $\dfrac{P \quad Q}{P \rightarrow Q}$

3. **Analyzing the use of matrix in different aspects**
   - Sparsity
   - Differentiability
   - Using complex numbers



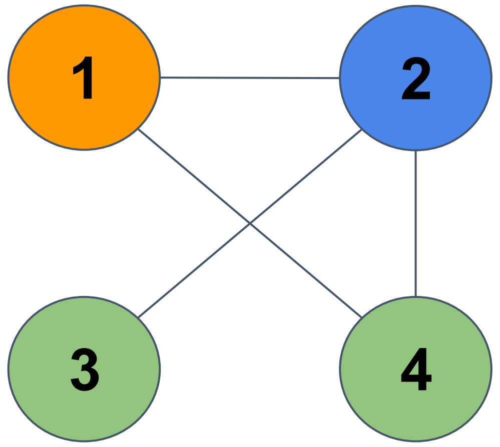
# Differentiable SAT Solver in Vector Spaces

**Koji Watanabe**

(5-Year Doctoral Student)

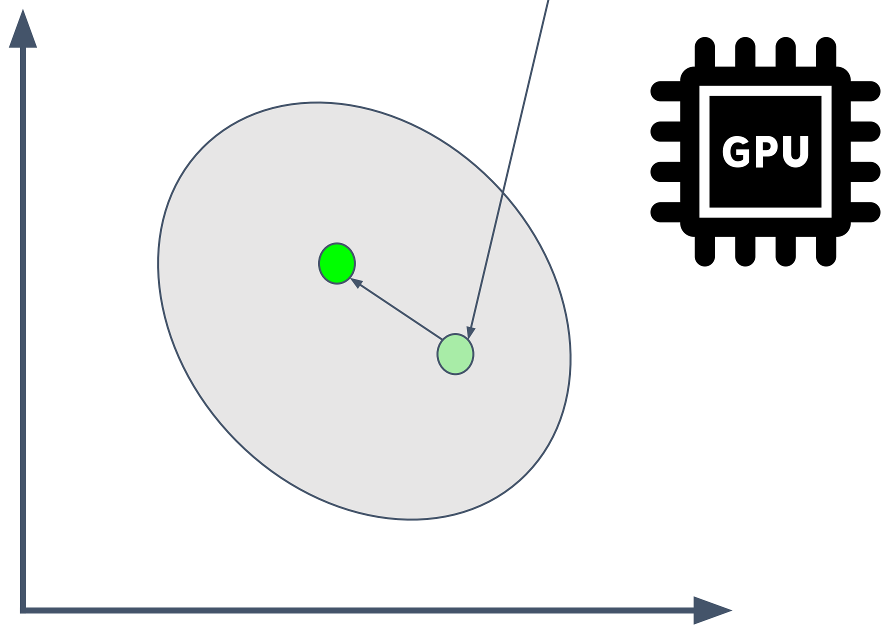
## Real World

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | | | 7 | | | | |
| 6 | | | 1 | 9 | 5 | | | |
| | 9 | 8 | | | | | 6 | |
| 8 | | | | 6 | | | | 3 |
| 4 | | | 8 | | 3 | | | 1 |
| 7 | | | | 2 | | | | 6 |
| | 6 | | | | | 2 | 8 | |
| | | | 4 | 1 | 9 | | | 5 |
| | | | | 8 | | | 7 | 9 |




## Symbolic Spaces

$$
\begin{aligned}
S = \ & (x_1 \vee x_2, \vee \neg x_3 \ldots \vee x_n) \\
& \wedge (x_1 \vee x_2, \vee x_3 \ldots \vee \neg x_n) \\
& \quad \vdots \\
& \wedge (\neg x_1 \vee x_2, \vee x_3 \ldots \vee x_n)
\end{aligned}
$$




## Vector Spaces

Finding a solution that satisfies a logical equation is called **SAT problem**, and the software for solving the SAT problem is called **SAT solver**. This study proposes **a new differentiable SAT solver** that maps the symbolic problem SAT to a vector space and finds the solution minimizing a cost function. Replacing symbolic representations with **vector and tensor representations**, we aim to improve the **speed and scalability** by using hardware specialized for parallel computing such as GPUs.

NII OPEN HOUSE 2021

大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

# Plan & Goal Recognition in Real Time Strategy Games

**Guillaume Lorthioir**
(Doctoral student)

Input: set of possible goals
Output: most likely goal and plan

Set of possible goals

$\{g_1, \dots, g_n\}$

Online planner

$I \xrightarrow{P_1} g_1$

$\vdots$

$I \xrightarrow{P_n} g_n$

New observation $O_i$

$P_1$ satisfies the observation: nothing is done

$\vdots$

$P_n$ does not satisfy the observation: $P_n =$

$I \xrightarrow{P_n^1} O_i \xrightarrow{P_n^2} g_n$

Pruning

$P_1$ does not satisfy pruning criteria, we keep $P_1$ and $g_1$

$\vdots$

$P_n$ satisfies pruning criteria, we remove $P_n$ and $g_n$

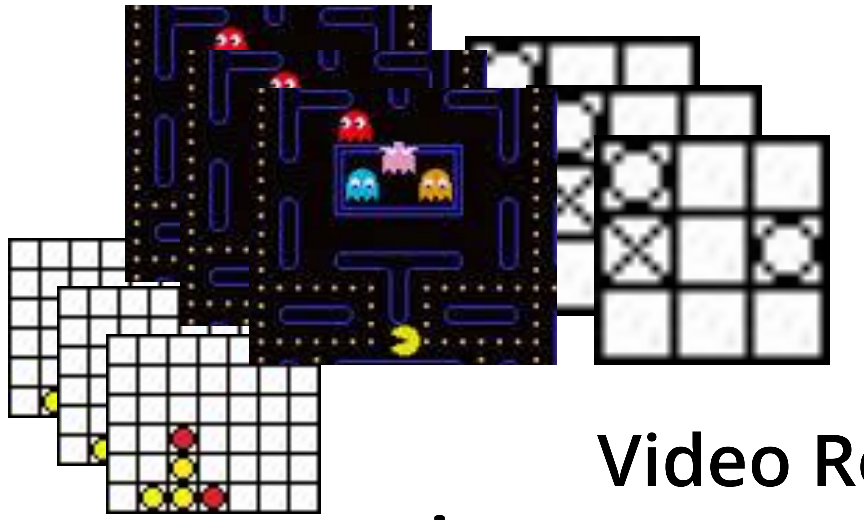
REPEAT

大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

Trying to **predict the plan and goal of the opponent player** in a game of StarCraft in real-time. For each possible goal for the player, we design one plan to achieve it, we modify these plans according to **partial observations of the player's action**, and then we estimate the most likely plan and goal.

# Extract Game Rules from Game Video Records

**Zheheng Xu**
(Master Student)



Video Records

+

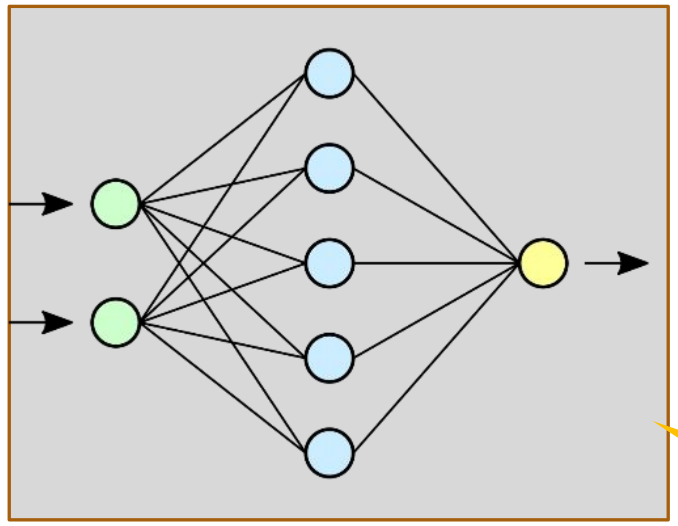
Action Records

Genetic Algorithm

010001001010

Interpretable Transition Data

Machine Learning



Deep Convolutional Neural Network

**Learning from Interpretation Transition (LFIT)**

Inductive Logic Programming

**Generative Adversarial Networks (GANs) abstract the videos**

block_1_1_blank -: block_1_1_blank
block_1_1_p1 -: block_1_1_p1
block_1_1_p1 -: block_1_1_ blank, 1_1_p1
block_1_1_p2 -: block_1_1_p2
block_1_1_p2 -: block_1_1_ blank, 1_1_p1
block_1_2_blank -: block_1_2_blank
block_1_2_p1 -: block_1_2_p1
block_1_2_p1 -: block_1_2_ blank, 1_1_p1
block_1_2_p2 -: block_1_2_p2
block_1_2_p2 -: block_1_2_ blank, 1_1_p1
block_1_3_blank -: block_1_3_blank
block_1_3_p1 -: block_1_3_p1
block_1_3_p1 -: block_1_3_ blank, 1_1_p1
……

Logic Programs



**Understand games, Explain games, Learn from games.**

# Neural-Symbolic Learning and Reasoning

**Yin Jun Phua**
(Doctoral Student)

## ∂LFIT+

**Input: Time series data**

ab → a → b → ...

**Output: NLP Encoded Matrix**

Rule body

Rule head

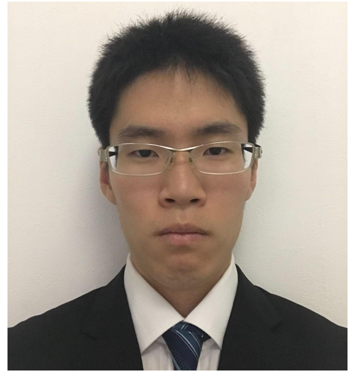
$$\begin{array}{cc} & \begin{array}{ccccccccc} \{\} & \{a\} & \{b\} & \{\neg a\} & \{\neg b\} & \{a,b\} & \{\neg a,b\} & \{a,\neg b\} & \{\neg a,\neg b\} \end{array} \\ \begin{array}{c} a \\ b \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

We interact with various **complex dynamic systems** everyday. These systems with composed of components evolve over time. Understanding the influences between these components provide valuable insights. **LFIT** is an unsupervised learning algorithm which learns the dynamics just by observing state transitions. LFIT has been mostly implemented in symbolic methods that utilize logical operations to induce logic programs. In our work, **we combine neural network with this previously symbolic algorithm.** We proposed a general model **∂LFIT** that learns to predict logic programs that explain the underlying observations. However, this model suffered from a combinatorial explosion problem. To solve the scalability issue, we propose a new model **∂LFIT+** that is capable of learning systems that are larger than previously possible.

Learning Logic Programs from Noisy State Transition Data [Phua *et al.*, ILP 2019]

▼[Link to research article]

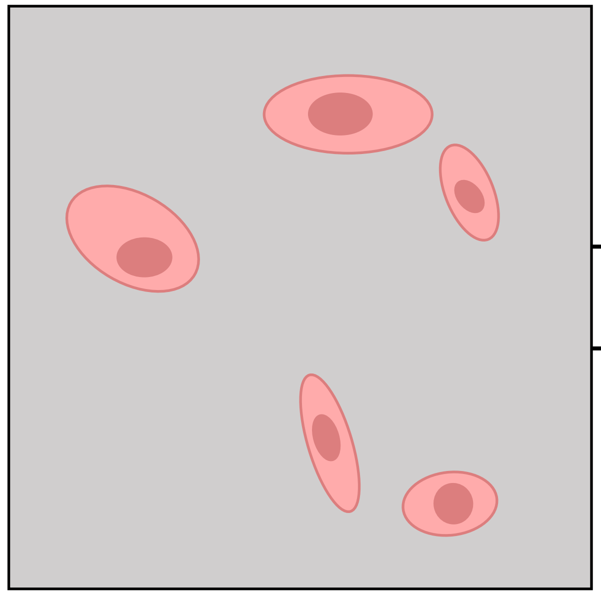
# Towards Learning Live-cell Dynamics




Ryota Yakushiji
(Master Student)


## 放射線下における細胞挙動の理解に向けた、機械学習による細胞増殖過程の予測モデル構築

がん治療などへの応用可能性



放射線下細胞増殖イメージング
(光学顕微鏡動画；５日間撮影)

→ **前処理**
特徴抽出
探索的データ解析

→
- 放射線量
- 生存細胞数
- 死滅細胞数
- 細胞分裂発生数
- 細胞融合発生数

→ **学習**
解釈遷移からの学習

→ 細胞数予測値

解釈遷移からの学習（Learning from Interpretation Transition; LFIT）[Inoue *et al.*, MLJ 2014]
観測時系列から動的システムの背後に潜む法則を学習。標準論理プログラムで表現。

▼Link to research article

# Equation Discovery in SARS-CoV-2 Network Dynamics

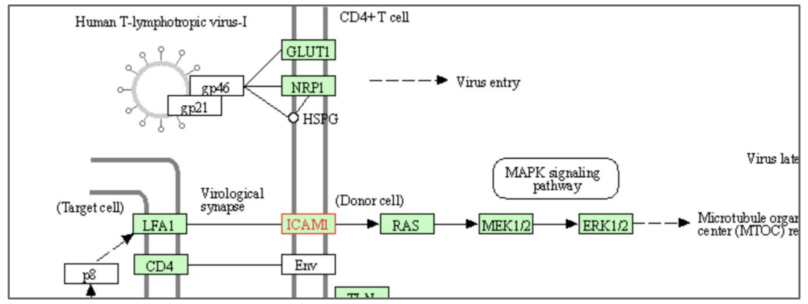
## ~Robust to Uncertainty, Interpretable to Causality~


**Mitsuhiro Odaka**
(5-Year Doctoral Student)

Mitsuhiro ODAKA

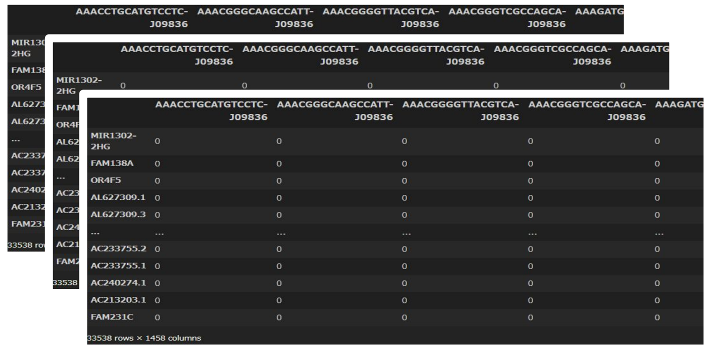
Google Scholar researchmap


SARS-CoV-2 *in vivo* dynamical systems




KEGG Pathway

Time-series scRNA-seq



**Learning**

**SR NA**

SR: Symbolic regression
NA: Numerical analysis

## Equations (Eqs)

(ODEs, PDEs, DAEs, S-system, ...)

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{N} X_j^{g_{i,j}} - \beta_i \prod_{j=1}^{N} X_j^{h_{i,j}} \quad (i = 1, \cdots, N)$$

## Logic programs (LPs)

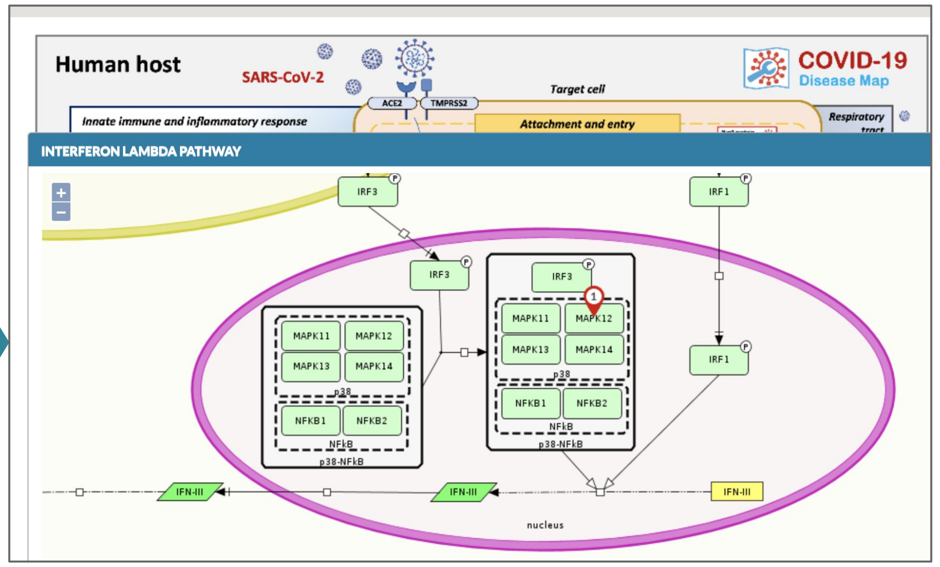
(QDEs, TCTL, ...)

ADD(X,Y,Z)
MULT(X,Y,Z)
DERIV(X,Y)
MINUS(X,Y)
$M^{+}$(X,Y)

**Reasoning**

**MoC**

MoC: Model checking




Reconstructive networks

My research attempts to build a new framework for **discovering the governing equations and hypotheses of complex system dynamics in viral infection**. This framework **combines dynamical systems theory, machine learning, and symbolic reasoning**. In other words, by organically linking differential equations, logic programs, time-series observations, and background knowledge, I aim to develop a tool for finding models **robust to uncertainty or perturbation and interpretable to causality**. One challenging task is **"multiscale modeling,"** which considers the crosstalk between systems of different biological hierarchies. Applications of such frameworks could include other viruses and real-world scenarios. One particularly urgent and significant focus is the **novel coronavirus infection (COVID-19)**. While working on the proposed framework, I am currently addressing case studies such as **completion and reconstruction of the COVID-19 Disease Map**, a developing knowledge graph on the pathogenic virus (SARS-CoV-2), and **dynamic network biomarker identification**.