# Inductive Logic Programming (ILP)

**Takeru Isobe**

**SOKENDAI**

September 24, 2024

# Table of contents

# Why inductive logic programming (ILP)?

Inductive logic programming [Muggleton,1991] is a **logical** form of machine learning.

- Interpretable hypothesis

  ex) $grandparent(V_1, V_2) \leftarrow parent(V_1, V_3), parent(V_3, V_2).$

- Learn from prior knowledge

  ex) $mother(a, b),\ parent(V_1, V_2) \leftarrow mother(V_1, V_2)$

- Exploit logical means

  ex) logical constraint, inverse entailment

[Muggleton,1991] Stephen H. Muggleton. Inductive Logic Programming. New Gener. Comput., 8(4):295–318, 1991.

# General issues on ILP

ILP systems have struggled with the following difficulties.

- Lack of expressive power
- Too large searchable space
- Noise in training examples or background knowledge (BK)
- Irrelevant BK

# Learning settings

There are several different learning settings.

- Learning from entailment (LFE)
- Learning from interpretation (LFI) [De Raedt,1997]
- Learning from transition (LFT) [Inoue,2014]

[De Raedt,1997] De Raedt, L. Logical settings for concept-learning. Artif. Intell., 95(1), 187–201, 1997.

[Inoue,2014] Inoue, K., Ribeiro, T., Sakama, C. Learning from interpretation transition. Mach Learn 94, 51–79, 2014.

# Learning from entailment: a typical learning setting

Learning from entailment (LFE) is a typical learning formalization of ILP.

**Definition**

Suppose background knowledge $B$, positive examples and negative examples$E^+$, $E^-$. the goal is to find a hypothesis $H$ such that

$$B \wedge H \models E^+,\ B \wedge H \not\models E^- \ \textit{and}\ B \wedge H \ \textit{is consistent.}$$

**Examples**

$B = \{mother(a, b)\}, E^+ = \{parent(a, b)\}, E^- = \{parent(b, a)\},$

$$H = \{parent(v1, v2) \leftarrow mother(v1, v2).\}$$

# Language of hypothesis and BK

Several different subsets of logic programs can be considered.

- Definite logic programs
- Normal logic programs
- Full clausal logic programs
- Datalog programs
- Answer set programs

# Expressive hypotheses

Some expressive ILP systems can learn the following hypotheses.

- Recursive hypothesis

**Examples**

$ancestor(V_1, V_2) \leftarrow parent(V_1, V_2).$
$ancestor(V_1, V_2) \leftarrow parent(V_1, V_2), ancestor(V_3, V_2).$

- Hypotheses with predicate invention (PI)

**Examples**

$inv(V_1, V_2) \leftarrow mother(V_1, V_2).$
$inv(V_1, V_2) \leftarrow father(V_1, V_2).$
$grandparent(V_1, V_2) \leftarrow inv(V_1, V_3), inv(V_3, V_2).$

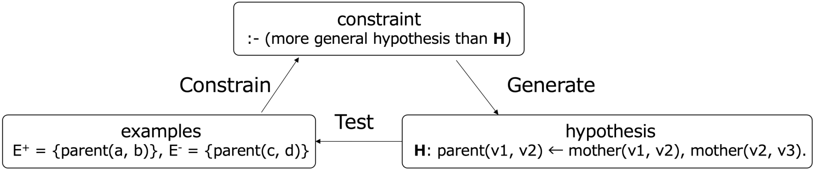
Note that the predicate "inv" represents the relation of parent.

# Popper: learning from failures

Learning from failures (LFF) [Cropper,2021] uses generality order about hypotheses to prune hypothesis space of LFE.

**Examples**

$H_1 = parent(V_1, V_2) \leftarrow mother(V_1, V_2).$
$H_2 = parent(V_1, V_2) \leftarrow mother(V_1, V_2), mother(V_2, V_3).$
$H_1$ is more general than $H_2$. ($H_2$ is more specific than $H_1$.)




[Cropper,2021] Andrew Cropper and Rolf Morel. Learning programs by learning from failures. Mach. Learn., 110(4):801–856, 2021.