Nonmonotonic Inductive Logic Programming

Chiaki Sakama
Wakayama University, Japan

Invited Talk at LPNMR 2001

Nonmonotonic Logic Programming (NMLP): Historical Background

- Negation as Failure (Clark, 1978)
- Stratified Negation (Apt et al., 1988, etc)
- Stable Model Semantics (Gelfond and Lifschitz, 1988)
- Well-founded Semantics (Van Gelder et al., 1988)
- Answer Set Semantics (Gelfond and Lifschitz, 1990), ...
- LPNMR conferences (1991 ～)

Inductive Logic Programming (ILP): Historical Background

- Least Generalization (Plotkin, 1970)
- Model Inference System (Shapiro, 1981)
- Inverse Resolution (Muggleton and Buntine, 1988)
- Inductive Logic Programming (Muggleton, 1990), ...
- ILP conferences (1991～)

NMLP vs. ILP: Goals

- NMLP: Representing incomplete knowledge and reasoning with commonsense.
- ILP: Inductive construction of first-order clausal theories from examples and background knowledge.

NMLP and ILP have seemingly different goals, but they have much in common in the background of problems.

NMLP vs. ILP: Common Background

- Discovering human knowledge is the iteration of hypotheses generation and revision, which is inherently nonmonotonic.
- Induction problems assume background knowledge which is incomplete, otherwise there is no need to learn.

➤ Representing and reasoning with incomplete knowledge are essential in ILP.

NMLP vs. ILP: Common Background

- Hypotheses generation in NMLP is done by abductive LP. Abduction and induction are both inverse deduction and extend theories to account for evidences.
- Updating general rules in NMLP is captured as the problem of concept-learning in ILP.

➤ NMLP and ILP handle the problems of hypotheses generation and knowledge update in different contexts.

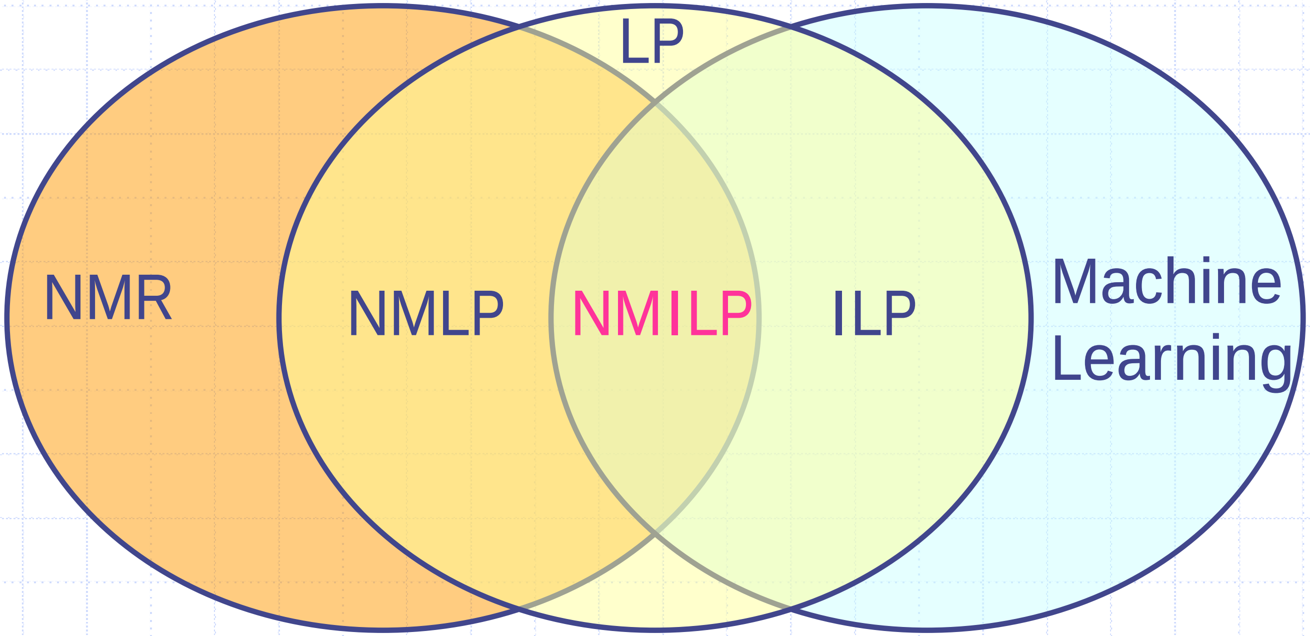
NMLP vs. ILP: Current Techniques

	NMLP	ILP
Inference	Default Reasoning Abduction, etc	Inductive Learning
Representation Language	LP with NAF (NLP, EDP, etc)	Clausal Theory (Horn LP, PDP)

- The present NMLP and ILP have limitations and complement each other.

NMILP: Combination of NMLP and ILP

- Since both commonsense reasoning and machine learning are indispensable for realizing AI systems, combining two fields is meaningful and important.



NMILP: Perspectives

- How can NMLP influence ILP ?
  - Extend the representation language
  - Use well-established theoretical or procedural tools in NMLP
- How can ILP influence NMLP ?
  - Introduce learning mechanisms to programs
  - Open new applications of NMLP

NMILP: What is the problem?

- Techniques in ILP have been centered on clausal logic, especially Horn logic.
- NMLP is different from classical logic, then existing techniques of ILP are not directly applicable to NMILP.
- Extension of the present framework and reconstruction of theories of ILP are necessary.

Contents of This Talk

- NMLP and ILP: Overview
- Induction in Nonmonotonic LP
  - Least Generalization
  - Inverse Resolution
  - Inverse Entailment
  - Other Techniques
- Open Issues

Normal Logic Programs (NLP)

- An NLP is a set of rules of the form:

  $A \leftarrow B_1, \ldots, B_m, \text{not } B_{m+1}, \ldots, \text{not } B_n$

  (A, B_i: atom, not B_j : NAF-literal)
- An LP-literal is either an atom or an NAF-literal.
- An interpretation I satisfies a rule R if

  $\{B_1, \ldots, B_m\} \subseteq I$ and $\{B_{m+1}, \ldots, B_n\} \cap I = \phi$

  imply $A \in I$ (written $I \models R$).

Stable Model Semantics

- An NLP is consistent if it has a stable model.
- An NLP is categorical if it has a single stable model.
- If every stable model of a program P satisfies a rule R, it is written as $P \models_s R$. Else if no stable model satisfies R, it is written as $P \models_s \text{not } R$.
- The expansion set of a stable model M is $M^+ = M \cup \{ \text{not } A \mid A \in HB \setminus M \}$ where HB is the Herbrand base.

Induction Problem

Given:

- a background knowledge B
- a set E^+ of positive examples
- a set E^- of negative examples

Find: hypotheses H satisfying

1. $B \cup H \models e$ for every $e \in E^+$
2. $B \cup H \not\models f$ for every $f \in E^-$
3. $B \cup H$ is consistent

Induction Algorithm

- An induction algorithm is correct if every hypothesis produced by it satisfies the conditions 1-3.
- An induction algorithm is complete if it produces every hypothesis satisfying 1-3.

- The goal of ILP is to develop a correct (and complete) algorithm which efficiently computes hypotheses.

Impractical Aspect on Completeness

B: $r(f(x)) \leftarrow r(x)$, $q(a) \leftarrow$, $r(b) \leftarrow$.

E: $p(a)$.

H: $p(x) \leftarrow q(x)$,

$p(x) \leftarrow q(x), r(b)$,

$p(x) \leftarrow q(x), r(f(b))$, ...

- Infinite number of hypotheses exist in general (and many of them are useless)
- Extracting meaningful hypotheses is most important (cf. induction bias)

NMILP Problem

Given:

- a background knowledge B as NLP
- a set E^+ of positive examples
- a set E^- of negative examples

Find: hypotheses H satisfying

1. $B \cup H \models_s e$ for every $e \in E^+$
2. $B \cup H \not\models_s f$ for every $f \in E^-$
3. $B \cup H$ is consistent under the stable model semantics

Subsumption and Generality Relation

- Given two clauses C_1 and C_2, C_1 θ-subsumes C_2 if $C_1\theta \subseteq C_2$ for some θ.
- Given two rules R_1 and R_2, R_1 θ-subsumes R_2 (written $R_1 \geqq_\theta R_2$) if $head(R_1)\theta = head(R_2)$ and $body(R_1)\theta \subseteq body(R_2)$ hold for some θ.

In this case, R_1 is said more general than R_2 under θ-subsumption.

Least Generalization under Subsumption

- Let R be a finite set of rules s.t. every rule in R has the same predicate in the head. A rule R is a least generalization of R under θ -subsumption if

  (i) $R \geqq_{\theta} R_i$ for every rule R_i in R , and
  (ii) $R' \geqq_{\theta} R$ for any other rule R' satisfying $R' \geqq_{\theta} R_i$ for every R_i in R .

Example

$R_1 = p(a) \leftarrow q(a), \text{not } r(a),$

$R_2 = p(b) \leftarrow q(b), \text{not } r(b).$

$R = p(x) \leftarrow q(x), \text{not } r(x).$

$R' = p(x) \leftarrow q(x).$

$R \geqq_\theta R_1$, $R \geqq_\theta R_2$, $R' \geqq_\theta R_1$, $R' \geqq_\theta R_2$, $R' \geqq_\theta R$.

R is a least generalization of R_1 and R_2.

Existence of Least Generalization

Theorem 3.1

Let R be a finite set of rules s.t. every rule in R has the same predicate in the head. Then every non-empty set R⊆R has a least generalization under θ-subsumption.

Computation of Least Generalization (1)

- A least generalization of two terms $f(t_1,\ldots,t_n)$ and $g(s_1,\ldots,s_n)$ is:

$$\begin{cases} \text{a new variable } v & \text{if } f \neq g; \\ f(lg(t_1,s_1), \ldots, lg(t_n,s_n)) & \text{if } f = g \end{cases}$$

- A least gen. of two LP-literals L_1 and L_2 is:

$$\begin{cases} \text{undefined if } L_1 \text{ and } L_2 \text{ do not have the same predicate and sign;} \\ lg(L_1,L_2) = (not)p(lg(t_1,s_1), \ldots, lg(t_n,s_n)), \text{ otherwise, where } L_1=(not)p(t_1,\ldots,t_n) \text{ and } L_2=(not)p(s_1,\ldots,s_n) \end{cases}$$

Computation of Least Generalization (2)

- A least generalization of two rules $R_1 = A_1 \leftarrow \Gamma_1$ and $R_2 = A_2 \leftarrow \Gamma_2$, where A_1 and A_2 have the same predicate, is obtained by

$$lg(A_1, A_2) \leftarrow \Gamma$$

where $\Gamma = \{\ lg(\gamma_1, \gamma_2) \mid \gamma_1 \in \Gamma_1, \gamma_2 \in \Gamma_2$ and $lg(\gamma_1, \gamma_2)$ is defined $\}$.

Relative Subsumption between Rules

- Let P be an NLP, and R_1 and R_2 be two rules. Then, R_1 θ-subsumes R_2 relative to P (written $R_1 \geqq^{P}_{\theta} R_2$) if there is a rule R that is obtained by unfolding R_1 in P and R θ-subsumes R_2.

  In this case, R_1 is said more general than R_2 relative to P under θ-subsumption.

Example

P: has_wing(x) $\leftarrow$ bird(x), not ab(x),
bird(x) $\leftarrow$ sparrow(x),
ab(x) $\leftarrow$ broken_wing(x).

R_1: flies(x) $\leftarrow$ has_wing(x).

R_2: flies(x) $\leftarrow$ sparrow(x), full_grown(x), not ab(x).

From P and R_1,

R_3: flies(x) $\leftarrow$ sparrow(x), not ab(x)

is obtained by unfolding.

Since R_3 θ -subsumes R_2, $R_1 \geqq^{P}_{\theta} R_2$.

Existence of Relative Least Generalization

- Least generalization does not always exist under relative subsumption.
- Let P be a finite set of ground atoms, and R_1 and R_2 be two rules.

Then, a least generalization of R_1 and R_2 under relative subsumption is computed as a least generalization of R_1' and R_2' where $head(R_i') = head(R_i)$ and $body(R_i') = body(R_i) \cup P$.

Example

P : bird(tweety) $\leftarrow$, bird(polly) $\leftarrow$.

R_1: flies(tweety) $\leftarrow$ has_wing(tweety),
not ab(tweety)

R_2: flies(polly) $\leftarrow$ sparrow(polly), not ab(polly)

Incorporating the facts from P into the bodies of R_1 and R_2:

Example (cont.)

R_1': flies(tweety) ← bird(tweety), bird(polly),
has_wing(tweety), not ab(tweety)

R_2': flies(polly) ← bird(tweety), bird(polly),
sparrow(polly), not ab(polly)

Computing least generalization of R_1' and R_2':

flies(x) ← bird(tweety), bird(polly),
bird(x), not ab(x).

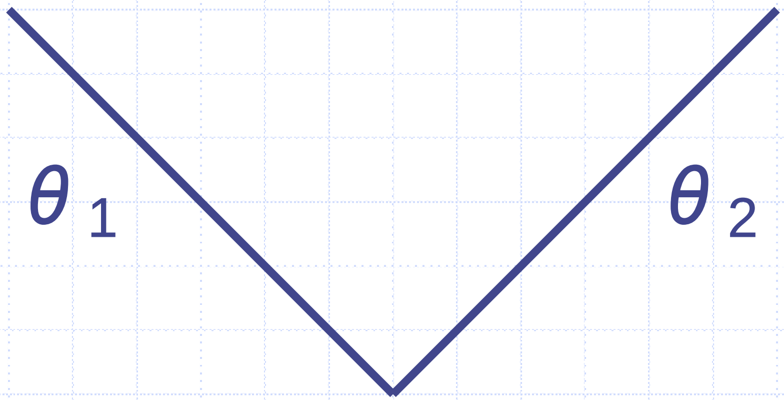
Removing redundant literals,

flies(x) ← bird(x), not ab(x).

Inverse Resolution

$R_1: B_1 \leftarrow \Gamma_1$ $\qquad$ $R_2: A_2 \leftarrow B_2, \ \Gamma_2$

θ_1 $\qquad$ θ_2 $\qquad$ $B_1\theta_1 = B_2\theta_2$



$R_3: A_2\theta_2 \leftarrow \Gamma_1\theta_1, \Gamma_2\theta_2$

Absorption constructs R_2 from R_1 and R_3.
Identification constructs R_1 from R_2 and R_3.
These are called V-operators.

V-operators

- Given NLP P containing R_1 and R_3, absorption produces

  $$A(P)=(P\setminus\{R_3\}) \cup \{R_2\}.$$

- Given NLP P containing R_2 and R_3, identification produces

  $$I(P)=(P\setminus\{R_3\}) \cup \{R_1\}.$$

- V(P) means either A(P) or I(P).
- $V(P) \models P$ if P is a Horn program.

V-operators do not generalize nonmonotonic programs

P: $p(x) \leftarrow \text{not } q(x),\ q(x) \leftarrow r(x),$
$s(x) \leftarrow r(x),\ s(a) \leftarrow .$

A(P): $p(x) \leftarrow \text{not } q(x),\ q(x) \leftarrow s(x),$
$s(x) \leftarrow r(x),\ s(a) \leftarrow .$

- ☑ $P \models_s p(a)$ but $A(P) \not\models_s p(a)$

V-operators may make a program inconsistent

P: p(x)← q(x), not p(x), q(x)← r(x),
s(x)← r(x), s(a)←.

A(P): p(x)← q(x), not p(x), q(x)← s(x),
s(x)← r(x), s(a)←.

- P has a stable model, but A(P) has no stable model.

V-operators may destroy the syntactic structure of programs

P: $p \leftarrow q$, $r \leftarrow q$, $r \leftarrow not\ p$.

A(P): $p \leftarrow r$, $r \leftarrow q$, $r \leftarrow not\ p$.

- P is acyclic and locally stratified, but A(P) is neither acyclic nor locally stratified.

Conditions for the V-operators to Generalize Programs

Theorem 3.2 (Sakama, 1999)

Let P be an NLP, $R_1 = B_1 \leftarrow \Gamma_1$, $R_2 = A_2 \leftarrow B_2, \Gamma_2$, and R_3 the resolvent of R_1 and R_2.

For any NAF-literal not L in P,

(i) if L does not depend on the head of R_3, then $P \models_s N$ implies $A(P) \models_s N$ for any $N \in HB$.

(ii) if L does not depend on B_2 of R_2, then $P \models_s N$ implies $I(P) \models_s N$ for any $N \in HB$.

Example

P: $flies(x) \leftarrow sparrow(x), \text{not } ab(x),$

$bird(x) \leftarrow sparrow(x),$

$sparrow(tweety) \leftarrow, \quad bird(polly) \leftarrow .$

E: flies(polly).

where $P \not\models_s flies(polly)$. By absorption,

A(P): $flies(x) \leftarrow bird(x), \text{not } ab(x),$

$bird(x) \leftarrow sparrow(x),$

$sparrow(tweety) \leftarrow, \quad bird(polly) \leftarrow .$

Then, $A(P) \models_s flies(polly)$.

Inverse Entailment

Suppose an induction problem

$$B \cup \{ H \} \models E$$

where B is a Horn program and H and E are each single Horn clause.

By inverting the entailment relation,

$$B \wedge \{ \neg E \} \models \neg H \ .$$

A possible hypothesis H is then deductively constructed by B and E.

Problems of IE in Nonmonotonic Programs

In NML deduction theorem does not hold in general [Shoham, 1987].

- This is due to the difference between the entailment relation $\models$ in classical logic and the relation $\models_{NML}$ in NML.

Ex. $P=\{ a \leftarrow \text{not } b\}$ has the stable model $\{a\}$, then $P \models_s a \leftarrow b$.

By contrast, $P \cup \{b\}$ has the stable model $\{b\}$, so $P \cup \{b\} \not\models_s a$.

Nested Rule

- A nested rule is defined as

  $A \leftarrow R$

  where A is an atom and R is a rule.
- An interpretation I satisfies a ground nested rule $A \leftarrow R$ if $I \models R$ implies $A \in I$.
- For an NLP P, $P \models_s (A \leftarrow R)$ if $A \leftarrow R$ is satisfied in every stable model of P.

Entailment Theorem

Theorem 3.3 (Sakama, 2000)

Let P be an NLP and R a rule s.t. $P \cup \{R\}$ is consistent. For any ground atom A, $P \cup \{R\} \models_s A$ implies $P \models_s A \leftarrow R$.

In converse, $P \models_s A \leftarrow R$ and $P \models_s R$ imply $P \cup \{R\} \models_s A$.

- The entailment theorem corresponds to the deduction theorem and is used for inverting entailment in NLP.

Inverse Entailment in NLP

Theorem 3.4 Let P be an NLP and R a rule s.t. $P \cup \{R\}$ is consistent. For any ground LP-literal L, if $P \cup \{R\} \models_s L$ and $P \models_s \leftarrow L$, then $P \models_s not\ R$.

- The relation $P \models_s not\ R$ provides a necessary condition for computing a rule R satisfying $P \cup \{R\} \models_s L$ and $P \models_s \leftarrow L$.

Relevance

- Given two ground LP-literals L_1 and L_2, define $L_1 \sim L_2$ if L_1 and L_2 have the same predicate and contain the same constants.
- Let L be a ground LP-literal and S a set of ground LP-literals. Then,
  L_1 in S is relevant to L if either (i) $L_1 \sim L$ or (ii) L_1 shares a constant with L_2 in S s.t. L_2 is relevant to L.

Computing Hypotheses by IE

- Suppose a function-free and categorical program P s.t. $P \cup \{R\} \models_s A$ and $P \models_s \leftarrow A$ for a ground atom A.
- By $P \models_s not\ R$,

  $M \not\models R$

  for the stable model M of P.
- Consider $\leftarrow \Gamma$ where Γ consists of ground LP-literals in M^+ which are relevant to A.

Computing Hypotheses by IE

Since M does not satisfy this constraint,

$$M \not\models \leftarrow \Gamma.$$

By $P \models_s \leftarrow A$, $A \notin M$, thereby not $A \in M^+$.

Since not A is relevant to A, not A is in $\leftarrow \Gamma$.

Shifting A to the head produces

$$A \leftarrow \Gamma'$$

where $\Gamma' = \Gamma \setminus \{\text{not } A\}$.

Finally, construct a rule R* s.t.

$$R^*\theta = A \leftarrow \Gamma' \quad \text{for some } \theta.$$

Correctness Theorem

Theorem 3.6 (Sakama, 2000)

Let P be a function-free and categorical NLP, A a ground atom, and R* a rule obtained as above. If $P \cup \{R^*\}$ is consistent and the predicate of A does not appear in P, then $P \cup \{R^*\} \models_s A$.

Example

P: $\text{bird}(x) \leftarrow \text{penguin}(x)$,

$\text{bird}(tweety) \leftarrow$, $\text{penguin}(polly) \leftarrow$.

L: flies(tweety).

Initially, $P \models_s \leftarrow \text{flies}(tweety)$ holds.

Our goal is to construct a rule R satisfying

$P \cup \{R\} \models_s L$.

First, P has the stable model M:

$M = \{\text{bird}(t), \text{bird}(p), \text{penguin}(p)\}$.

Example (cont.)

- From M, the expansion set M^+ is
- $M^+ = \{$ bird(t), bird(p), penguin(p), not penguin(t), not flies(t), not flies(p) $\}$
- Picking up LP-literals relevant to L:
  - $\leftarrow$ bird(t), not penguin(t), not flies(t).
- Shifting L=flies(t) to the head:
  - flies(t) $\leftarrow$ bird(t), not penguin(t).
- Replacing tweety by a variable x:
  - flies(x) $\leftarrow$ bird(x), not penguin(x).

Other Algorithms (1): Learning from NLP

- Ordinary ILP + Specialization
  (Bain and Muggleton, 1992)
- Selection from candidate hypotheses
  (Bergadano et al., 1996)
- Generic top-down ILP
  (Martin and Vrain, 1996; Seitzer, 1995; Fogel and Zaverucha, 1998)

Other Algorithms (2): Learning from ELP

- Ordinary ILP + Prioritization

  (Dimopoulos and Kakas, 1992)
- Ordinary ILP + Specialization

  (Inoue and Kudoh, 1997; Lamma et al., 2000)
- Learning by Answer Sets

  (Sakama, 2001)

Open Issues (1)

- Generalization under implication
  - In ILP generalization is defined by two different orders; the subsumption order and the implication order.
  - Under the implication order, C_1 is more general than C_2 if $C_1 \models C_2$.
  - In NMLP the entailment relation is defined under the commonsense semantics, so that generalization under implication would have effect different from Horn ILP.

Open Issues (2)

- Generalization operations in NMLP
  - In clausal theories, operations by inverting resolution generalize programs, but they do not generalize NMLP in general.
  - Program transformations which generalize NMLP in general (under particular semantics) are unknown.
  - Such operations would serve as basic operations in NMILP.

Open Issues (3)

- Relations between induction and other commonsense reasoning
  - Induction is a kind of NMR. Is there any theoretical relations between induction and other nonmonotonic formalisms?
  - Such relations enable us to implement ILP in terms of NMLP, and open possibilities to integrate induction and other commonsense reasoning.

Final Remark

- 10 years have past since the 1st LPNMR in 1991. In LPNMR'91, the preface says:

...there has been growing interest in the relationship between LP semantics and NMR. It is now reasonably clear that there is ample scope for each of these areas to contribute to the other.

Now the sentence is rephrased to combine NMLP and ILP in the context of NMILP.