KRR with Uncertainty

Inference in LP^{MLN}

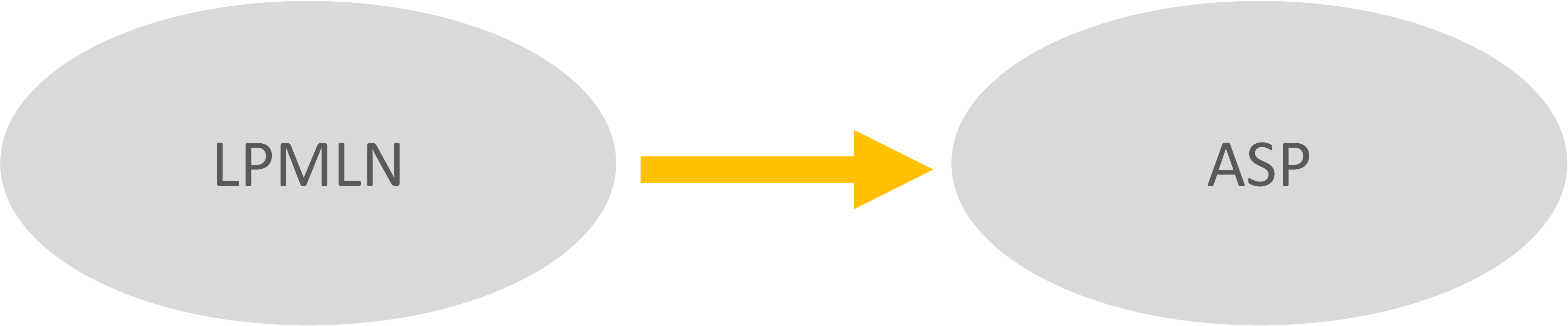
Ira A. Fulton Schools of Engineering
Arizona State University

Objectives

Objective

Apply LPMLN inference method on probabilistic reasoning

From LPMLN to ASP: Weak Constraints



Weak Constraints (1 of 3)

- A weak constraint has the form

  :~ F. [Weight @ Level]

where F is a conjunction of literals (atoms and their negations)

- **Weight** is an integer and **Level** is a nonnegative integer
  - When F is satisfied, we say the weak constraint is violated, and we apply a penalty of Weight at this Level — the higher level it is, the higher priority we consider

Weak Constraints (2 of 3)

- Let Π be a program $\Pi_1 \cup \Pi_2$, where Π_1 is a usual ASP program and Π_2 is a set of weak constraints.

- We call I a stable model of Π if it is a stable model of Π_1 .

- For every stable model I of Π and any nonnegative integer l, the penalty of I at level L, denoted by $Penalty_\Pi(I, L)$, is defined as

$$\sum_{\substack{:\sim\ F[w@l]\in\Pi_2,\\ I\models F}} w.$$

- ex:

```
{p; q}.
:~ p.   [10@0]
:~ q.   [5@1]
```

Pen({p}, 0) = 10
Pen({p}, 1) = 0

Weak Constraints (3 of 3)

- For any two stable models I and I' of Π, we say I is **dominated** by I' if
  - there is some level L such that $Penalty_{\Pi}(I', L) < Penalty_{\Pi}(I, L)$ and
  - for all integers $K > L$, $Penalty_{\Pi}(I', K) = Penalty_{\Pi}(I, K)$
- A stable model of Π is called **optimal** if it is not dominated by another stable model of Π

In clingo

```
% test

{p;q}.
:~ p. [10@0]
:~ q. [5@1]
```

```
$ clingo test

Answer: 1
Optimization: 0 0
OPTIMUM FOUND


Models        : 1
  Optimum     : yes
Optimization  : 0 0
```

```
$ clingo test --opt-mode=enum 0
Solving...
Answer: 1
                          level 1  level 0

Optimization: 0 0
Answer: 2
q
Optimization: 5 0
Answer: 3
p
Optimization: 0 10
Answer: 4
p q
Optimization: 5 10
OPTIMUM FOUND


Models          : 4
```

Translation lpmln2asp

For soft rules

$$w_i : Head_i \leftarrow Body_i$$

$$unsat(i) \leftarrow Body_i,\ not\ Head_i$$

$$Head_i \leftarrow Body_i, not\ unsat(i)$$

$$:\sim unsat(i)\ [w_i]$$

- when $Body_i$ is true but $Head_i$ is false, we say this rule is unsatisfied;
- when the rule is satisfied, we include this rule in deriving the stable models;
- when this rule is unsatisfied, the stable model receives a penalty w_i .

Translation lpmln2asp

Soft Rules:

$$w_i : Head_i \leftarrow Body_i$$

$$unsat(i) \leftarrow Body_i, not\ Head_i$$
$$Head_i \leftarrow Body_i, not\ unsat(i)$$
$$:\sim unsat(i)\ [w_i@0]$$

Hard Rules:

$$\alpha : Head_i \leftarrow Body_i$$

$$unsat(i) \leftarrow Body_i, not\ Head_i$$
$$Head_i \leftarrow Body_i, not\ unsat(i)$$
$$:\sim unsat(i)\ [1@1]$$

Theorem: For any LP^{MLN} program Π, the most probable stable models of Π are precisely the optimal stable models of $\mathrm{lpmln2asp}(\Pi)$.

Example

Theorem: For any LP^{MLN} program Π, the most probable stable models of Π are precisely the optimal stable models of lpmln2asp(Π).

LP^{MLN} program

$$\alpha : p \qquad (r_1)$$

$$10 : q \leftarrow p \qquad (r_2)$$

$$-20 : q \qquad (r_3)$$

Q: What is the most probable stable model?

I	$W(I)$
$\emptyset$	e^{10}
$\{p\}$	e^{α}
$\{q\}$	e^{-10}
$\{p, q\}$	$e^{\alpha-10}$

Example

LP^{MLN} program	Clingo program	Clingo Output
$\alpha : p \quad (r_1)$	$unsat(1) \;:-\; not\;\; p.$ $p :-not\; unsat(1).$ $:\sim unsat(1).\; [1@1]$	
$10 : q \leftarrow p \quad (r_2)$	$unsat(2) \;:-p, not\;\; q.$ $q :-p, not\; unsat(2).$ $:\sim unsat(2)\; [10@0]$	Solving… Answer: 1 p unsat(2) unsat(3) Optimization: 0 -10
$-20 : q \quad (r_3)$	$unsat(3) \;:-not\;\; q.$ $q :-\, not\; unsat(3).$ $:\sim unsat(3). [-20@0]$	OPTIMUM FOUND % The number in blue is the penalty at level 1. % The number in red is the penalty at level 0.

Implementation of LPMLN2ASP

The most probable stable models correspond to optimal stable models

Weight of stable models can be calculated with

$$W_{\Pi}^{\text{pnt}}(I) = exp\Bigg(- \sum_{\mathtt{unsat}(i, w_i, \mathbf{c}) \in \phi(I)} w_i \Bigg).$$

The corresponding stable model of the corresponding ASP program lpmln2asp(Π)

Marginal probability of an atom a

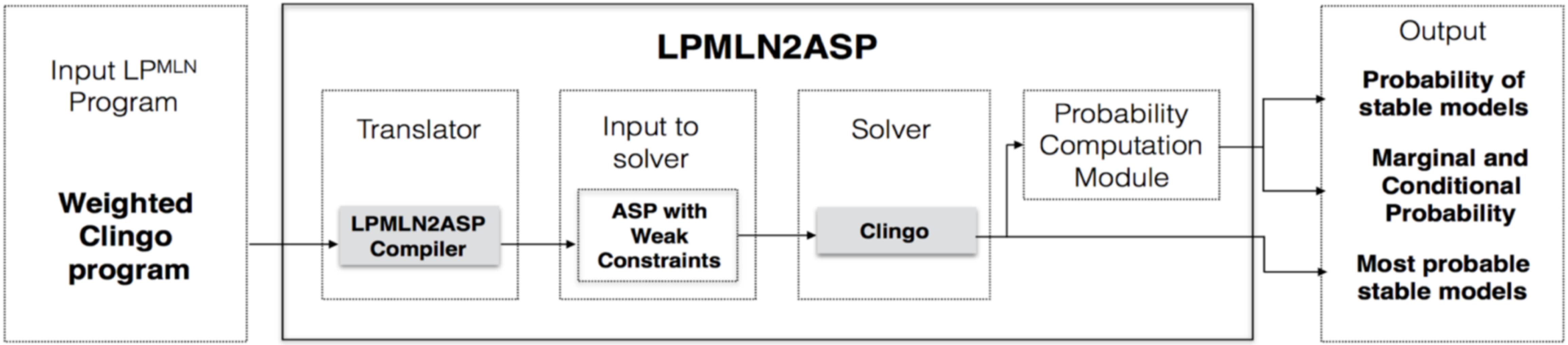
$$P_{\Pi}(a) = \sum_{J \models a} P_{\Pi}(J)$$

Conditional probability of an atom a given evidence E

$$P_{\Pi}(a \mid E) = \sum_{J \models a} P_{\Pi \cup E}(J)$$

(E is encoded as a set of ASP constraints)

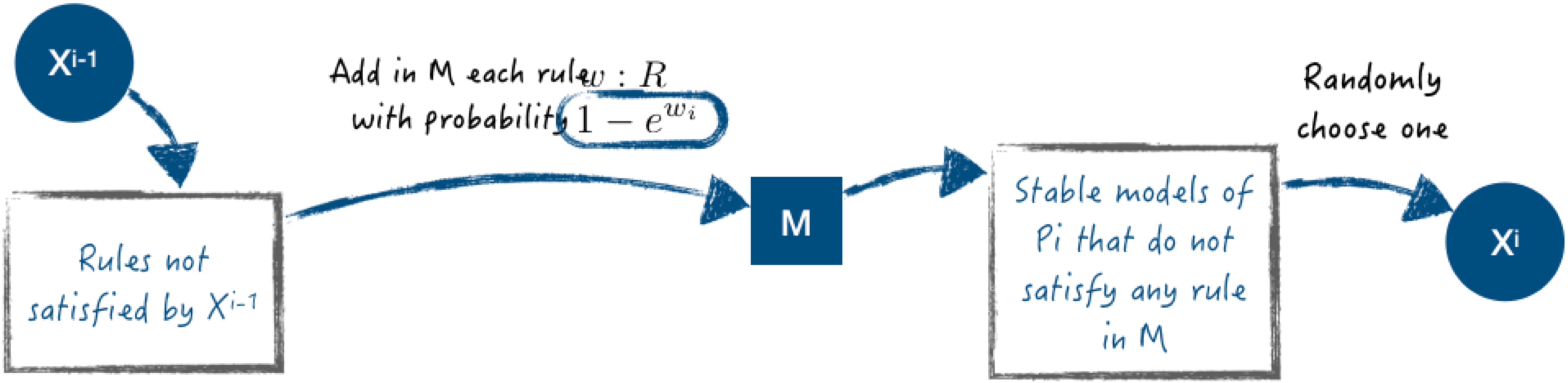
System Architecture



- lpmln2asp can compute MAP inference, marginal and conditional probability
- MAP inference is directly computed by clingo
- Probability calculations are computed by a probability computation module

Algorithm MC-ASP [Lee & Wang, 2018]

- Compute approximate probability by MCMC based sampling
- Adapted from MC-SAT for Markov Logic [Poon and Domingos, 2006]
- Start from a random probabilistic stable model
- Assume the weights are non-negative.
- Each sampling iteration:



Wrap-Up