Lifted MAP Inference
for Markov Logic

- Somdeb Sarkhel
- Deepak Venugopal
- Happy Mittal
- Parag Singla
- Vibhav Gogate

Outline

- Introduction
- Lifted MAP Inference for Markov Logic Networks
  - S. Sarkhel, D. Venugopal, P. Singla, V. Gogate – AISTATS'14
- An Integer Polynomial Programming Based Framework for Lifted MAP Inference
  - S. Sarkhel, D. Venugopal, P. Singla, V. Gogate – NIPS'14
- New Rules for Domain Independent Lifted MAP Inference
  - H. Mittal, P. Goyal, V. Gogate, P. Singla – NIPS'14

Outline

- **Introduction**
- Lifted MAP Inference for Markov Logic Networks
  - S. Sarkhel, D. Venugopal, P. Singla, V. Gogate – AISTATS'14
- An Integer Polynomial Programming Based Framework for Lifted MAP Inference
  - S. Sarkhel, D. Venugopal, P. Singla, V. Gogate – NIPS'14
- New Rules for Domain Independent Lifted MAP Inference
  - H. Mittal, P. Goyal, V. Gogate, P. Singla – NIPS'14

MAP Inference

- **Problem:** Find most likely state of world given evidence

$$\arg\max_{y} \; P(y \mid x)$$

Query

Evidence

MAP Inference

- **Problem:** Find most likely state of world given evidence

$$\arg\max_{y} \frac{1}{Z_x} \exp\left(\sum_i w_i n_i(x, y)\right)$$

MAP Inference

- **Problem:** Find most likely state of world given evidence

$$\arg\max_{y} \sum_{i} w_i n_i(x, y)$$

- Same as maximizing satisfied weight clause
- (Or minimizing unsatisfied weight clause)
- Alchemy solve using weighted SAT solver (e.g., MaxWalkSAT [Kautz et al., 1997])

Optimization Problems

minimize: $f_0(\mathbf{x})$ ← Objective Function

subject to:

$$f_i(\mathbf{x}) \leq 0$$
$$h_j(\mathbf{x}) = 0$$
$$\mathbf{u} \geq \mathbf{x} \geq \mathbf{0}$$
$$\mathbf{x} \in Z^n$$

Constraints

Range → $\mathbf{u} \geq \mathbf{x} \geq \mathbf{0}$

Type → $\mathbf{x} \in Z^n$

- Constraints are linear
- Variable type is integer

Integer Linear Programming

$$\begin{array}{ll} \textbf{minimize:} & \mathbf{c}^T\mathbf{x} \\ \textbf{subject to:} & \mathbf{Ax} \leq \mathbf{b} \\ & \mathbf{u} \geq \mathbf{x} \geq \mathbf{0} \\ & \mathbf{x} \in Z^n \end{array}$$

- Objective function is linear
- NP-Complete problem
- MAP can be solved using *ILP* solver (Used in RockIt [Noessner et al., 2013])

Outline

- *Introduction*
- **Lifted MAP Inference for Markov Logic Networks**
  - **S. Sarkhel, D. Venugopal, P. Singla, V. Gogate – AISTATS'14**
- An Integer Polynomial Programming Based Framework for Lifted MAP Inference
  - S. Sarkhel, D. Venugopal, P. Singla, V. Gogate – NIPS'14
- New Rules for Domain Independent Lifted MAP Inference
  - H. Mittal, P. Goyal, V. Gogate, P. Singla – NIPS'14

Lifted MAP Inference

- Lifted Algorithms …
- Group indistinguishable atoms together
  - ‘Super atom’
- Given assignment to these groups compute the weight (probability) of the assignment

Lifted MAP Inference

- For a subclass of MLNs – “*non-shared MLN*”
  - All possible assignments can be grouped by number of true groundings of its predicates

- *Non-shared MLN*: no logical variable is shared between the atoms in a formula.
  - Example: R(x) v S(y)

Lifted MAP Inference

R(A)	R(B)	S(A)	S(B)	Weight	Groups
0	0	0	0	0	(0,0)
0	0	0	1	$2w_1 + w_3$	(0,1)
0	0	1	0	$2w_1 + w_3$	(0,1)
0	0	1	1	$4w_1 + 2w_3$	(0,2)
0	1	0	0	$2w_1 + w_2$	(1,0)
0	1	0	1	$3w_1 + w_2 + w_3$	(1,1)
0	1	1	0	$3w_1 + w_2 + w_3$	(1,1)
0	1	1	1	$4w_1 + w_2 + 2w_3$	(1,2)
1	0	0	0	$2w_1 + w_2$	(1,0)
1	0	0	1	$3w_1 + w_2 + w_3$	(1,1)
1	0	1	0	$3w_1 + w_2 + w_3$	(1,1)
1	0	1	1	$4w_1 + 2w_3 + w_2$	(1,2)
1	1	0	0	$4w_1 + 2w_2$	(2,0)
1	1	0	1	$4w_1 + 2w_2 + w_3$	(2,1)
1	1	1	0	$4w_1 + 2w_2 + w_3$	(2,1)
1	1	1	1	$4w_1 + 2w_2 + 2w_3$	(2,2)

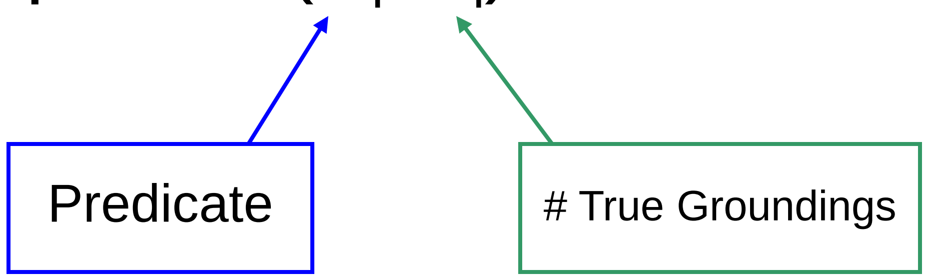
Group (0, 1)

Group (2, 1)

Figure 1: Weights of all assignments to ground atoms and (lifted) groups for the non-shared MLN: $[\mathrm{R}(x) \vee \mathrm{S}(y), w_1]$; $[\mathrm{R}(x), w_2]$; and $[\mathrm{S}(y), w_3]$ with domains given by $\Delta_x = \Delta_y = \{A, B\}$.

Lifted MAP Inference

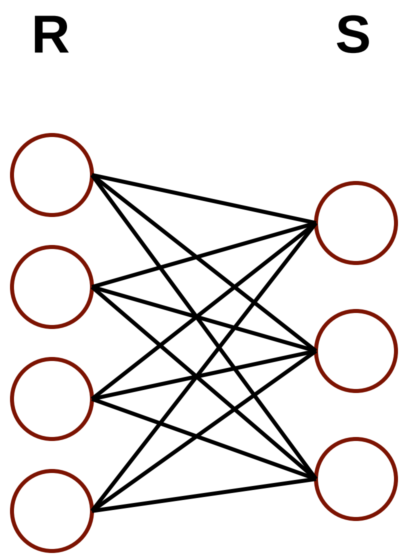
- All possible assignments of a non-shared MLN can be grouped by number of true groundings of its predicates
- **Counting Assignment:** ground assignments can be grouped as (R_i, a_i)



Lifted MAP Inference

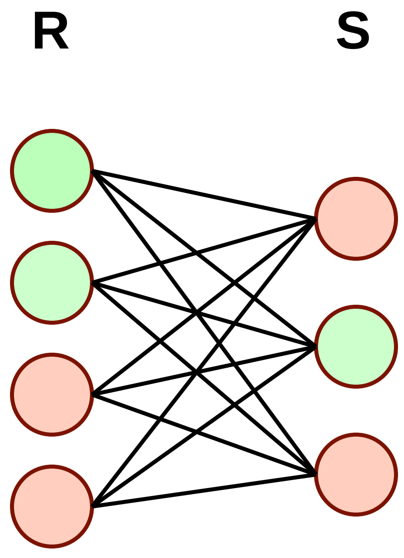
- Simple Lifted Algorithm: Iterate over all counting assignment of all predicates to find the one that has maximum weight
- Ground search space: $O(2^{nd})$
- Lifted search space: $O(d^n)$
- *Domain Lifted* (dependent on domain)
- Can we do better?
- Yes, in fact we can do much better!

Lifted MAP Inference

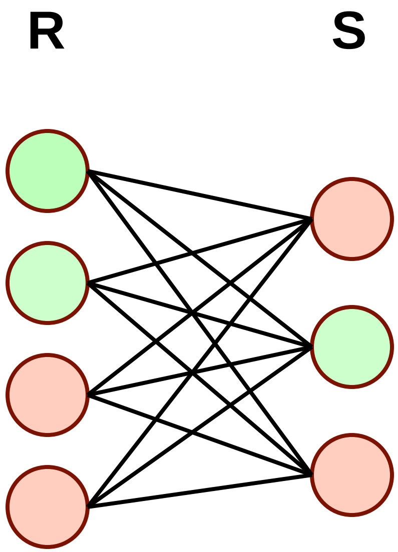


- Consider the MLN: R(x) v S(y), w_1
- $\Delta x = 4$, $\Delta y = 3$
- Let's compute weight for assignment [(R, 2), (S, 1)]

Lifted MAP Inference



Lifted MAP Inference



- No of unsatisfied clauses = 2×2
- Let's generalize for assignment $\omega = [(R, r), (S, s)]$
- $N(f, \omega) = \Delta R . \Delta S - (\Delta R - r)(\Delta S - s)$

Lifted MAP Inference

minimize: $\Sigma_i \Pi_j x_j$

subject to: $\mathbf{u} \geq \mathbf{x} \geq \mathbf{0}$

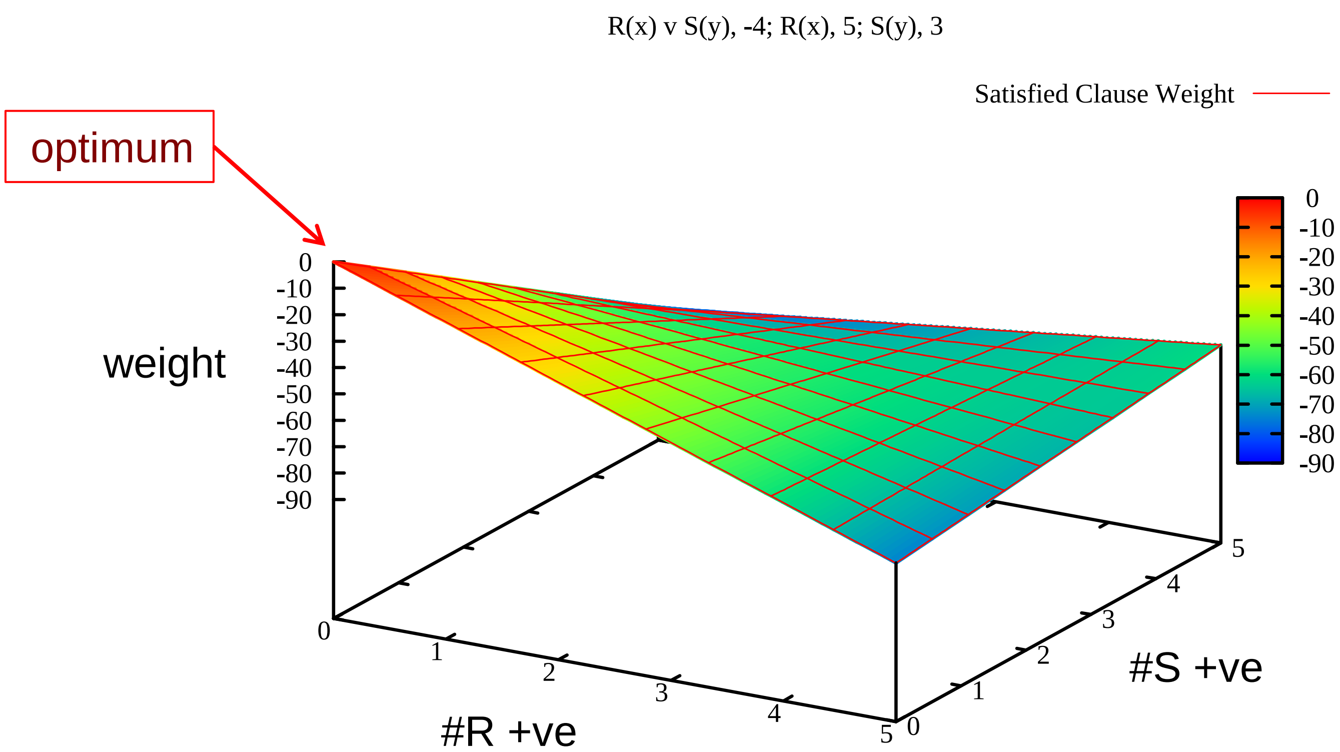
$\mathbf{x} \in Z^n$

No Constraints!

- Objective function is multilinear (i.e.- linear in each variable)
  - e.g. – $f(x_1,x_2,x_3) = x_1 + x_2x_3 – 2x_1x_3 + x_2 – x_1x_2x_3$
- Optimization problem has no constraint (except range and type)!

Lifted MAP Inference

- Non-shared MLN w/o self-joins

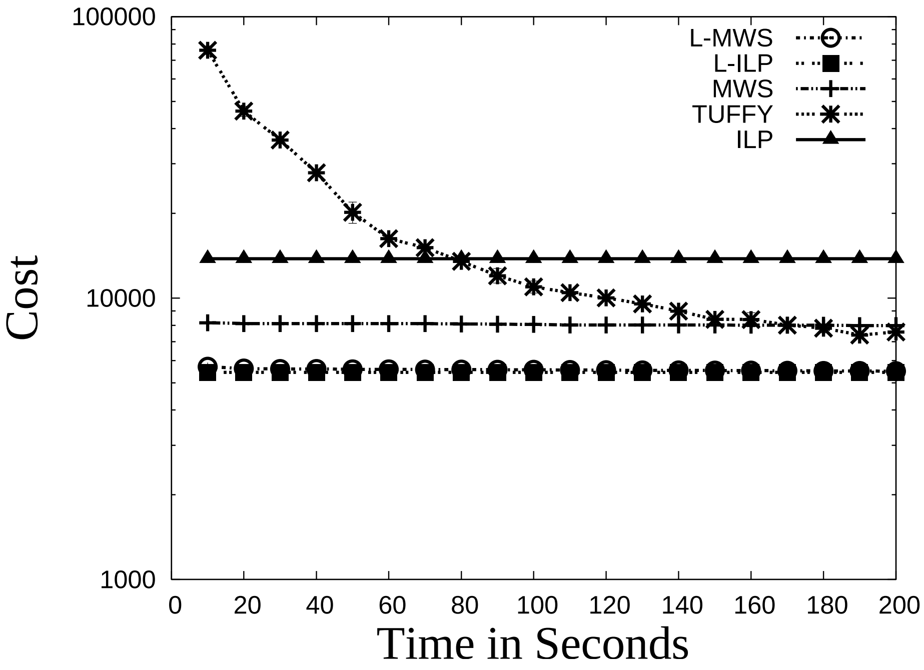


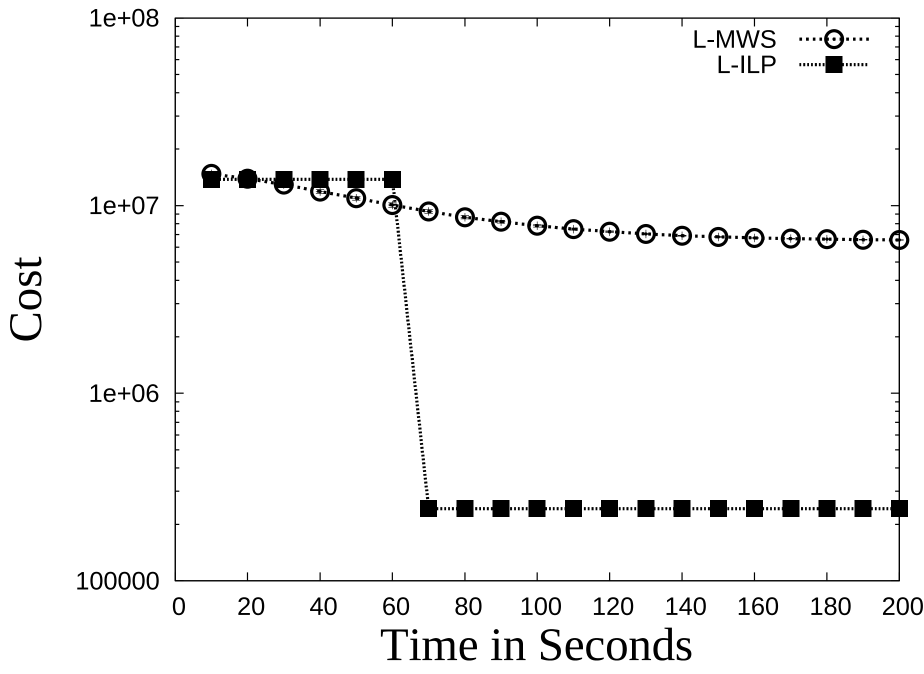
Satisfied clause weight as a surface plot of counting assignment

Lifted MAP Inference

- Non-shared MLN w/o self-joins
  - Optimum values lie on extreme points.
  - **Uniform Assignment**,
  - New Lifted Algorithm: Iterate over uniform assignment of all predicates to find the one that has maximum weight
  - Can be reduced to weighted SAT over predicates
  - Search Space: $O(2^n)$

Experiments





- Citation Information Extraction MLN

Lifted MAP Inference

- Contribution
  - Inference is *domain independent* and hence can scale for huge domains
  - No need to develop new solvers (use existing weighted SAT solver like WalkSAT or ILP based)
  - Inference time is guaranteed to be no worse than ground inference (no overhead)!

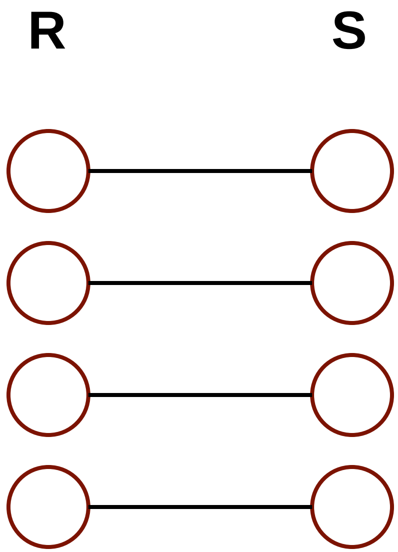
Outline

- Introduction
- Lifted MAP Inference for Markov Logic Networks
  - S. Sarkhel, D. Venugopal, P. Singla, V. Gogate – AISTATS'14
- **An Integer Polynomial Programming Based Framework for Lifted MAP Inference**
  - **S. Sarkhel, D. Venugopal, P. Singla, V. Gogate – NIPS'14**
- New Rules for Domain Independent Lifted MAP Inference
  - H. Mittal, P. Goyal, V. Gogate, P. Singla – NIPS'14

An Integer Polynomial Programming Based Framework for Lifted MAP Inference

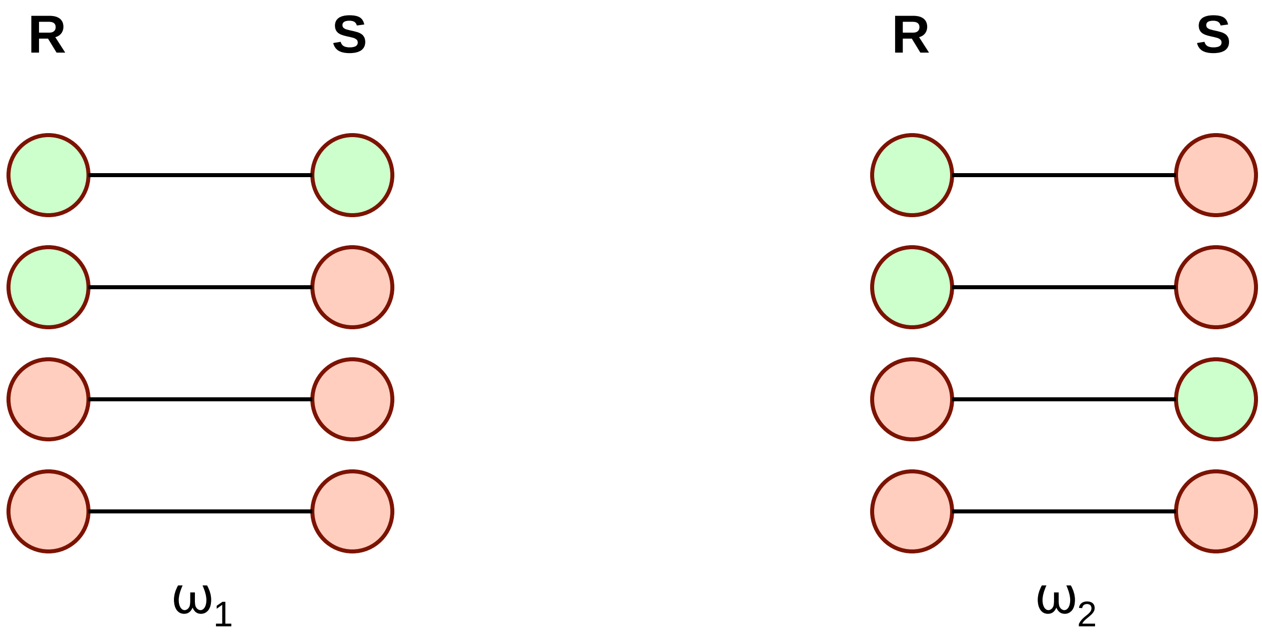
- Limitations of *Lifted MAP* [AISTATS'14]
  - Only applicable for formulas that have no shared terms.
  - Do not use many existing research on lifted inference (like the '*decomposer rule*')

An Integer Polynomial Programming Based Framework for Lifted MAP Inference



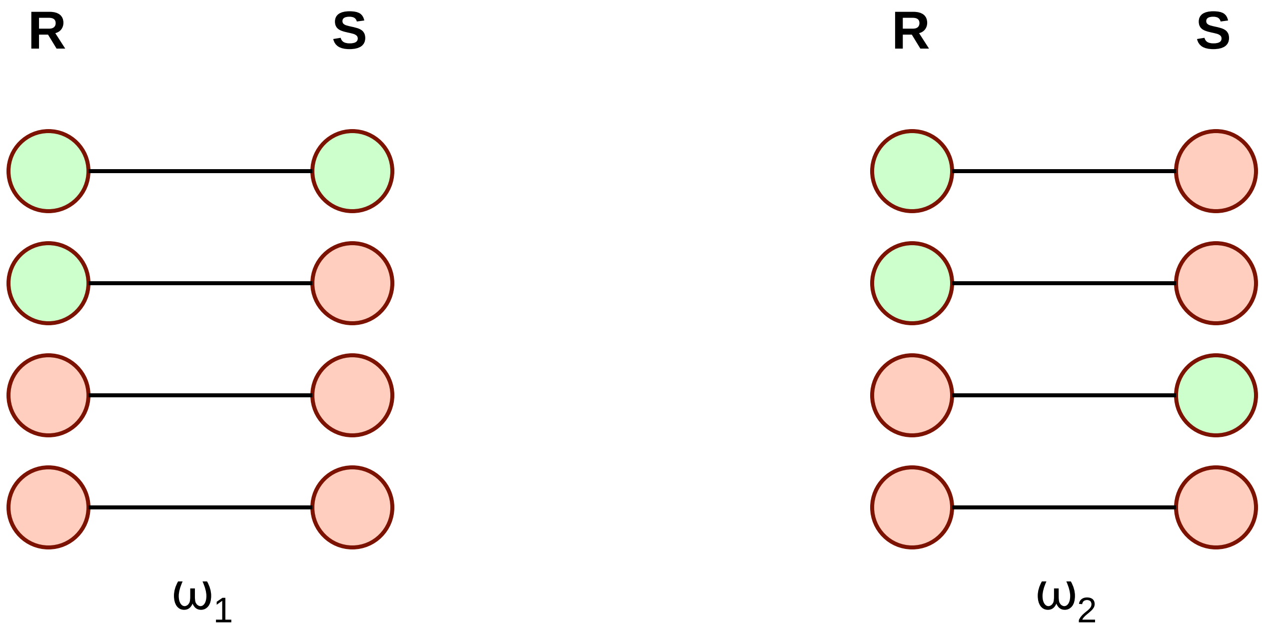
- Consider the MLN: R(x) v S(x), w_1
- $\Delta x = 4$
- Can we compute the weight for assignment [(R, 2), (S, 1)]

An Integer Polynomial Programming Based Framework for Lifted MAP Inference



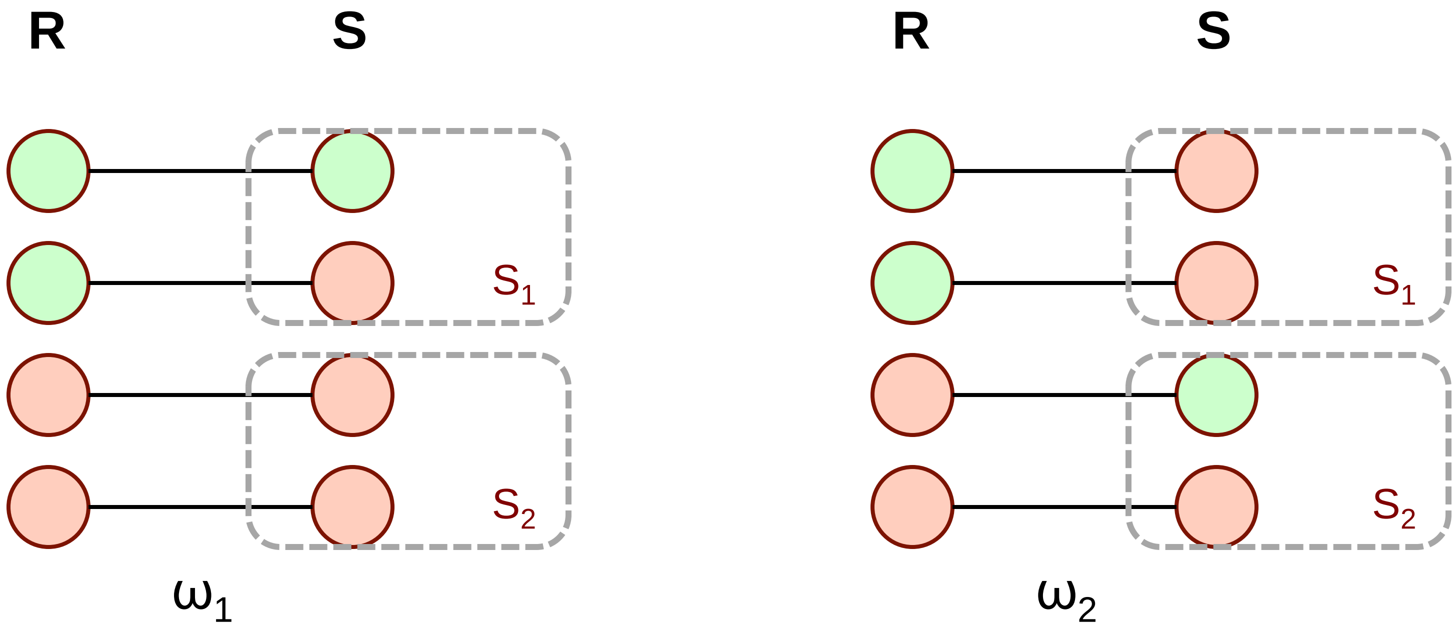
- $N(f, \omega_1) = N(f, \omega_2)$?
- No!

An Integer Polynomial Programming Based Framework for Lifted MAP Inference



- Solution = Shattering
- (S, *s*) is not sufficient. We need more info
- Which of the (S, *s*) joins with (R, *r*)

An Integer Polynomial Programming Based Framework for Lifted MAP Inference



[(R, 2), (S_1, 1) , (S_2, 0)]

[(R, 2), (S_1, 0) , (S_2, 1)]

An Integer Polynomial Programming Based Framework for Lifted MAP Inference

- Probabilistic theorem proving [1]
  - Advanced lifted inference technique
  - Extended for MAP Inference

- The weight of an assignment is calculated as:

$$w(\mathtt{A}, i) = \sum_{k=1}^{2} \sum_{f_j \in F(\mathtt{A}_k)} w_j \prod_{y \in V(f_j)} D(i_{\mathtt{R}}(y))$$

Algorithm 1 PTP-MAP(MLN M)

if M is empty **return** 0
Simplify(M)
if M has *disjoint* MLNs $M_1, \ldots, M_k$ **then**
  return $\sum_{i=1}^{k}$ PTP-MAP(M_i)
if M has a *decomposer* $\mathbf{d}$ such that $D(i \in \mathbf{d}) > 1$ **then**
  return PTP-MAP($M|\mathbf{d}$)
if M has an *isolated atom* $\mathtt{R}$ such that $D(i_{\mathtt{R}}) > 1$ **then**
  return PTP-MAP ($M|\{1_{\mathtt{R}}\}$)
if M has a *singleton atom* $\mathtt{A}$ **then**
  return $\max_{i=0}^{D(1_{\mathtt{A}})}$ PTP-MAP($M|(\mathtt{A}, i)) + w(\mathtt{A}, i)$
Heuristically select an argument $i_{\mathtt{R}}$
return PTP-MAP($M|G(i_{\mathtt{R}})$)

[1] V. Gogate and P. Domingos, *Probabilistic Theorem Proving*. UAI'11

An Integer Polynomial Programming Based Framework for Lifted MAP Inference

minimize:	$f_0(\mathbf{x})$
subject to:	$\mathbf{Ax} \leq \mathbf{b}$
	$\mathbf{u} \geq \mathbf{x} \geq \mathbf{0}$
	$\mathbf{x} \in Z^n$

- Objective function is polynomial in **x**
  - e.g. – $f(x_1,x_2,x_3) = x_1^2 + x_2x_3 - x_1x_3^2 + x_2 + 2x_1x_2x_3$

An Integer Polynomial Programming Based Framework for Lifted MAP Inference

- PTP-MAP performs an exhaustive search over all possible lifted assignments and can be slow
- To improve is encode as an IPP problem
- Algorithm 2 runs PTP-MAP schematically and forms the IPP
- Each variable corresponds to counting assignment

Algorithm 2 SMLN-2-IPP(SMLN S)

if S is empty **return** $\langle 0, \emptyset, \emptyset \rangle$
Simplify(S)
if S has disjoint SMLNs **then**
  for disjoint SMLNs $S_i \ldots S_k$ in S
    $\langle f_i, G_i, X_i \rangle$ = SMLN-2-IPP(S_i)
  return $\langle \sum_{i=1}^{k} f_i, \cup_{i=1}^{k} G_i, \cup_{i=1}^{k} X_i \rangle$
if S has a *decomposer* **d** **then**
  return SMLN-2-IPP($S|\mathbf{d}$)
if S has a *isolated singleton* R **then**
  return SMLN-2-IPP($S|\{i_{\mathtt{R}}\}$)
if S has a *singleton* atom A **then**
  Introduce an IPP variable 'i'
  Form a constraint g as '$(0 \leq i \leq D(1_{\mathtt{A}}))$'
  $\langle f, G, X \rangle$ = SMLN-2-IPP($S|(\mathtt{A}, i)$)
  return $\langle f + w(\mathtt{A}, i), G \cup \{g\}, X \cup \{i\} \rangle$
Heuristically select an argument $i_{\mathtt{R}}$
return SMLN-2-IPP($S|G(i_{\mathtt{R}})$)

An Integer Polynomial Programming Based Framework for Lifted MAP Inference

minimize: $f_0(\mathbf{x})$

subject to: $\mathbf{A}\mathbf{x} \leq \mathbf{b}$

$\mathbf{u} \geq \mathbf{x} \geq \mathbf{0}$

$\mathbf{x} \in Z^n$

- Constraints are linear
- Very general problem
- Not many off-the-shelf software present
- Difficult to solve!

An Integer Polynomial Programming Based Framework for Lifted MAP Inference

- Solving the IPP problem:
  - Solved by converting to Integer Linear Program
  - The classic method outlined in [2]
    - Convert IPP to 0-1 Polynomial Programming Problem
    - Replace each variable by binary sum expression
    - Simplify
    - Linearize it by adding additional variables
  - ILP can be solved using any ILP solver

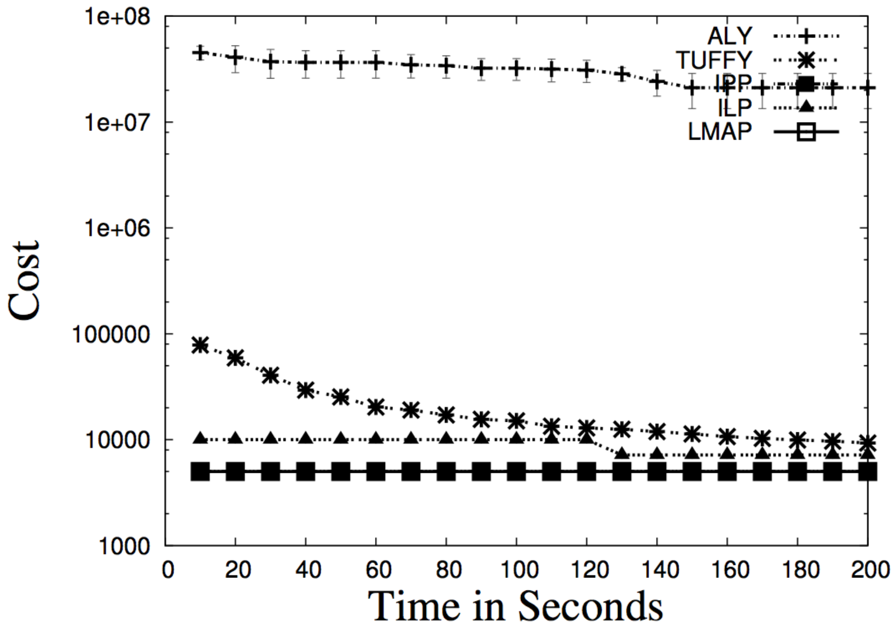
[2] Lawrence J Watters. Reduction of Integer Polynomial Programming Problems to Zero-One Linear Programming Problems. *Operations Research,* 15(6):1171–1174, 1967

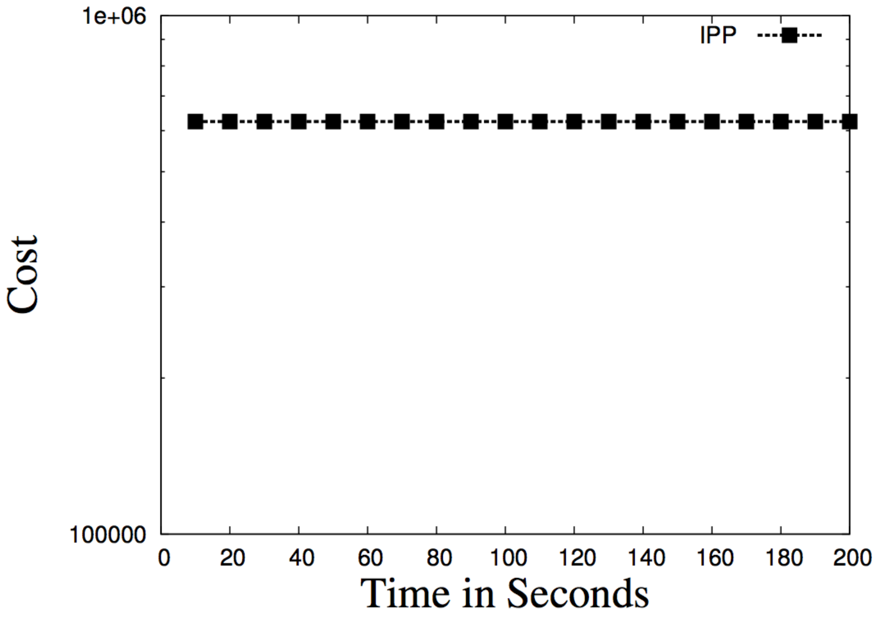
An Integer Polynomial Programming Based Framework for Lifted MAP Inference

- Solving the IPP problem:
  - Example
    - $f(x_1, x_2) = x_1 + x_2 - x_1x_2 \quad [0 \leq x_1,x_2 \leq 3]$
    - $x_1 = d_0^{(1)} + 2d_1^{(1)}, \qquad x_2 = d_0^{(2)} + 2d_1^{(2)}$
    - $f(d_0^{(1)},d_1^{(1)},d_0^{(2)},d_1^{(2)}) = d_0^{(1)} + 2d_1^{(1)} + d_0^{(2)} + 2d_1^{(2)} - (d_0^{(1)} + 2d_1^{(1)}) (d_0^{(2)} + 2d_1^{(2)})$
    - Replace $d_0^{(1)}.d_0^{(2)}$ by another variable d_{00} …
    - Add constraints – $d_0^{(1)} + d_0^{(2)} - d_{00} \leq 1$, $-d_0^{(1)} - d_0^{(2)} + 2d_{00} \leq 0$

Ensures $d_{00} = d_0^{(1)}.d_0^{(2)}$

Experiments





- Citation Information Extraction MLN

An Integer Polynomial Programming Based Framework for Lifted MAP Inference

- Contribution
  - Inference time (search space) is no worse than our AISTATS’14 approach
  - Applicable to any ‘*normal*’ MLN
  - Provides a framework to plug-in any newly discovered lifting rule

Questions?

Thank You!