Lifted Inference Rules With Constraints

Happy Mittal, Anuj Mahajan — Dept. of Comp. Sci. & Engg., IIT Delhi

Vibhav Gogate — Dept. of Comp. Sci., UT Dallas

Parag Singla — Dept. of Comp. Sci. & Engg., IIT Delhi

INTRODUCTION

Markov Logic Networks (MLN) [Richardson & Domingos, 2006]

- A Markov Logic Network (MLN) is a set of pairs (F,w) where
  - F is a formula in first order Logic
  - w is a real number (weight of formula)
- When a world violates a formula, it becomes less probable, but not impossible.
- Together with a set of constants, it defines a Markov network.

$\Delta_x = \Delta_y = \{1,2,\ldots,100\}$

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x,y\ Smokes(x) \wedge Friends(x,y) \Rightarrow Smokes(y)$

- **Joint Probability :**

$$P(y) = \frac{1}{Z}\exp\left(\sum_i w_i n_i(y)\right) \qquad Z = \sum_{assignment} \exp\left(\sum_i w_i n_i(y)\right)$$

Weight of Formula i — No. of true groundings of formula i — Intractable

- **Lifted Inference :** Exploit symmetry in the network and infer about group of objects together. Ex : Lifted VE, Lifted BP, Lifting rules.
- In above example, probability of smoking of every person is same.
- Suppose we have evidence : Smokes(1), Friends(1,2).
- Evidence breaks symmetry, and imposes constraints on the values variables can take.
- How to do lifted inference when constraints are imposed on variables ?
- Focus of our work :
  - Propose a new constraint language which supports efficient lifted inference.
  - Modify lifted inference rules to work with our proposed constraint language.

CONSTRAINT LANGUAGE (SetInEq)

- Weighted first order formulas with constraints.
- Three types of Atomic Constraints :
  - Subset Constraints: $w \quad P(x) \Rightarrow Q(y,z), x \in \{A,B\}$
  - Equality Constraints: $w \quad P(x) \Rightarrow Q(y,z), x = y$
  - Inequality Constraints: $w \quad P(x) \Rightarrow Q(y,z), x \neq y$
- Constraint Tuple : Conjunction of atomic constraints
  $w \quad P(x) \Rightarrow Q(y,z), x \in \{A,B\} \wedge y \notin \{C,D\} \wedge x \neq y$
- Constraint Set : Disjunction of constraint tuples
  $w \quad P(x) \Rightarrow Q(y,z), (x \in \{A,B\} \wedge x \neq y) \vee (y \notin \{C,D\})$

Operations

- **Join** : Conjunction of two constraint tuples .
  - linear in the size of constraint tuples being joined.
- **Project** : Eliminate a variable from a given constraint tuple.
  - may require splitting when inequality constraints are present.

  Ex : $x \neq y \wedge y \neq z$

Canonical Representation

- Only one subset constraint per variable in a constraint tuple.
  $x \in \{A,B\} \wedge x \in \{B,C\} \wedge y \notin \{C,D\}$ ⟹ $x \in \{B\} \wedge y \notin \{C,D\}$
- Identical supports for variables in eq/ineq constraints.
  $x \in \{A,B\} \wedge y \in \{A,B,C\} \wedge x = y$ ⟹ $x \in \{A,B\} \wedge y \in \{A,B\} \wedge x = y$

Splitting Operation: $x \in \{A,B,C\} \wedge y \in \{B,C,D\} \wedge x \neq y$ → $x \in \{A\} \wedge y \in \{B,C\}$; $x \in \{A\} \wedge y \in \{D\}$; $x \in \{B,C\} \wedge y \in \{B,C\} \wedge x \neq y$; $x \in \{B,C\} \wedge y \in \{D\}$

- Time Complexity to canonicalize a constraint set : $O(mk + k^3)$ m : no. of constants, k : no. of variables

LIFTING RULE 1 : DECOMPOSER RULE [Gogate & Domingos '11]

Decomposer Rule

$$Z\left(\boxed{w \quad P(x) \Rightarrow Q(x,y)}\right) \Longrightarrow \left(Z\left(\boxed{w \quad P(x) \Rightarrow Q(x,y)}\right)\right)^{10}$$

$\Delta_x = \Delta_y = \{1,2,\ldots,10\}$ ⟹ $\Delta_x = \{1\}, \Delta_y = \{1,2,\ldots,10\}$

$w \quad P(x) \Rightarrow Q(x,y)$ with $x \in \{1,2,3\} \wedge y \in \{1,2,3\} \vee x \in \{1,2\} \wedge y \in \{5,6\}$

Partitioning

Decomposer Rule With Constraints

$$Z\left(\begin{array}{c} w \quad P(x) \Rightarrow Q(x,y) \\ x \in \{1,2\} \wedge y \in \{1,2,3\} \\ \vee \\ x \in \{3\} \wedge y \in \{1,2,3\} \\ \vee \\ x \in \{1,2\} \wedge y \in \{5,6\} \end{array}\right) = \left(Z\left(\begin{array}{c} w \quad P(x) \Rightarrow Q(x,y) \\ x \in \{1\} \wedge y \in \{1,2,3\} \\ \vee \\ x \in \{1\} \wedge y \in \{5,6\} \end{array}\right)\right)^2 * \left(Z\left(\begin{array}{c} w \quad P(x) \Rightarrow Q(x,y) \\ x \in \{3\} \wedge y \in \{1,2,3\} \end{array}\right)\right)^1$$

LIFTING RULE 2 : BINOMIAL RULE [Gogate & Domingos '11]

Binomial Rule

MLN M: $w \quad P(x) \Rightarrow Q(x,y)$, $\Delta_x = \Delta_y = \{1,2,\ldots,10\}$

→ M_0: $w \quad True \Rightarrow Q^T(x,y)$; $w \quad False \Rightarrow Q^F(x,y)$

→ M_1: $w \quad True \Rightarrow Q^T(x,y)$; $w \quad False \Rightarrow Q^F(x,y)$

⋮

→ M_{10}: $w \quad True \Rightarrow Q^T(x,y)$; $w \quad False \Rightarrow Q^F(x,y)$

Consider theory M_k
$Q^T : \Delta_x = \{1,2,\ldots,k\}$
$Q^F, \Delta_x = \{k+1,\ldots,10\}$
$\Delta_y = \{1,2,\ldots,10\}$

$$Z(M) = \sum_{k=0}^{10} {}^nC_k * Z(M_k)$$

Binomial Rule With Constraints

$w \quad P(x) \Rightarrow Q(x,y)$, $x \in \{1,2,3\} \wedge y \in \{1,2,3\} \vee x \in \{1,2\} \wedge y \in \{5,6\}$

Partitioning

$w \quad P(x) \Rightarrow Q(x,y)$, $x \in \{1,2\} \wedge y \in \{1,2,3\} \vee x \in \{3\} \wedge y \in \{1,2,3\} \vee x \in \{1,2\} \wedge y \in \{5,6\}$

Applying Binomial Rule on P(x). Rule applied on {1,2}

- $w \quad True \Rightarrow Q(x,y)$, $[x \in \{\} \wedge y \in \{1,2,3\}] \vee [x \in \{\} \wedge y \in \{5,6\}]$
  $w \quad False \Rightarrow Q(x,y)$, $[x \in \{1,2\} \wedge y \in \{1,2,3\}] \vee [x \in \{1,2\} \wedge y \in \{5,6\}]$
  $w \quad P(x) \Rightarrow Q(x,y)$, $x \in \{3\} \wedge y \in \{1,2,3\}$
- $w \quad True \Rightarrow Q(x,y)$, $[x \in \{1\} \wedge y \in \{1,2,3\}] \vee [x \in \{1\} \wedge y \in \{5,6\}]$
  $w \quad False \Rightarrow Q(x,y)$, $[x \in \{2\} \wedge y \in \{1,2,3\}] \vee [x \in \{2\} \wedge y \in \{5,6\}]$
  $w \quad P(x) \Rightarrow Q(x,y)$, $x \in \{3\} \wedge y \in \{1,2,3\}$
- $w \quad True \Rightarrow Q(x,y)$, $[x \in \{1,2\} \wedge y \in \{1,2,3\}] \vee [x \in \{1,2\} \wedge y \in \{5,6\}]$
  $w \quad False \Rightarrow Q(x,y)$, $[x \in \{\} \wedge y \in \{1,2,3\}] \vee [x \in \{\} \wedge y \in \{5,6\}]$
  $w \quad P(x) \Rightarrow Q(x,y)$, $x \in \{3\} \wedge y \in \{1,2,3\}$

Restricted to x ϵ {1,2}

Restricted to x ϵ {3}

Apply Binomial Rule on {3}

EVIDENCE PROCESSING & NORMAL FORM

Normal Form : Unconstrained representation in which
(a) No constant in any formula
(b) Variables appearing in same predicate position have identical domains..

$w \quad P(x) \Rightarrow Q(y)$, $\Delta_x = \Delta_y = \{1,2,\ldots,n\}$

Evidence : $P(1) = P(2) = \ldots = P(n/2) = True$

SetInEq

$w \quad P(x) \Rightarrow Q(y)$, $x \in \{1,2,\ldots,n/2\} \wedge y \in \{1,2,\ldots,n\}$

$w \quad P(x) \Rightarrow Q(y)$, $x \in \{n/2+1,\ldots,n\} \wedge y \in \{1,2,\ldots,n\}$

Normal

$w \quad P_1(x) \Rightarrow Q(y)$, $\Delta_x = \{1\}, \Delta_y = \{1,2,\ldots,n\}$

⋮

$w \quad P_{n/2}(x) \Rightarrow Q(y)$, $\Delta_x = \{n/2\}, \Delta_y = \{1,2,\ldots,n\}$

$w \quad P'(x) \Rightarrow Q(y)$, $\Delta_x = \{n/2+1,\ldots,n\}, \Delta_y = \{1,2,\ldots,n\}$

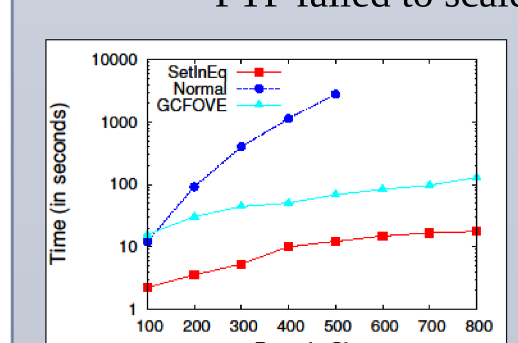
PREVIOUS APPROACHES

Approach	Constraint Type	Constraint Aggregation	Tractable Solver	Lifting Algorithm
Lifted VE [Poole 2003]	eq/ineq **no subset**	intersection **no union**	**No**	Lifted VE
CFOVE [Milch et al 2008]	eq/ineq **no subset**	intersection **no union**	Yes	Lifted VE
GCFOVE [Taghipour et al 2012]	Subset (tree based) **No inequality**	Intersection Union	Yes	Lifted VE
Approx LBP [Singla et al 2014]	Subset (hypercube) **No inequality**	Intersection Union	**No**	Lifted message passing
Knowledge Compilation (KC) [Broeck et al 2011]	eq/ineq subset	Intersection **no union**	**No**	First order knowledge compilation
Lifted Inference from other side [Jha et al 2010]	Normal forms **(no constraints)**	**None**	Yes	Lifting rules : decomposer, binomial etc
PTP [Gogate & Domingos 2011]	eq/ineq **no subset**	Intersection **no union**	no	lifted search & sampling: Decomposer, binomial
Current Work	eq/ineq subset	Intersection Union	Yes	lifted search & sampling: Decomposer, binomial, single occurrence

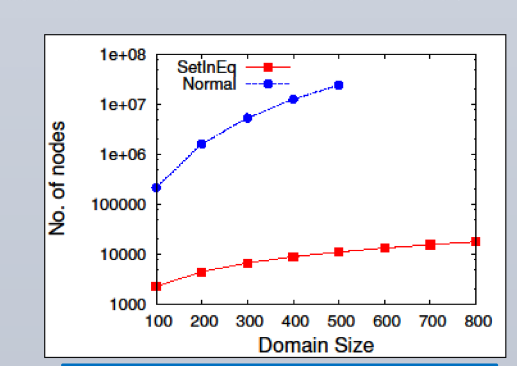
- SetInEq is one of the most expressive constraint languages..
- SetInEq allows for efficient constraint processing during lifting.

EXPERIMENTS & RESULTS

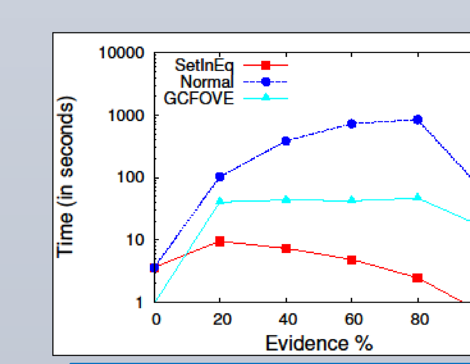
- Algorithms Compared :
  - SetInEq, Normal, GCFOVE, PTP
- Exact Inference
  - Performed exact inference on Friends and Smokers (FS) and Professors and students (PS) datasets.
  - PTP failed to scale to domain size of even 100.



FS : size vs time (sec)

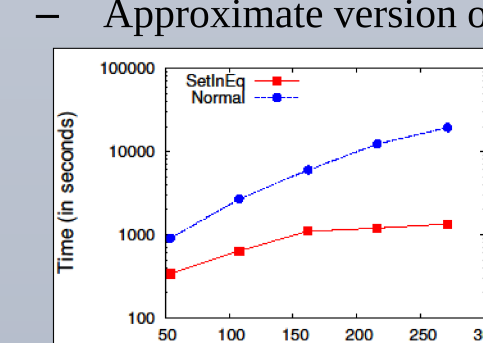


FS : size vs #nodes

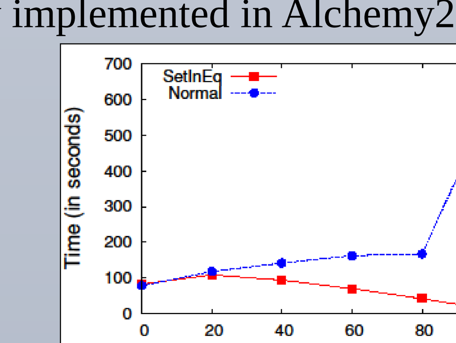


FS : evid% vs time (sec)

- Approximate Inference
  - Performed approximate inference on WebKB and IMDB datasets
  - GCFOVE doesn't have approximate version.
  - Approximate version of PTP is not fully implemented in Alchemy2



WebKB : size vs time (sec)



IMDB : evid% vs time (sec)