Lifted MAP Inference for Markov Logic Networks

Somdeb Sarkhel, Deepak Venugopal,
Parag Singla and Vibhav Gogate

(AISTATS2014)

詠み手: 井之上

Satisfiability Testing

- Input: set of clauses (tp. CNF)
  - e.g., $\{\neg rainy \vee wet, \neg sprinkler \vee wet, \neg rainy \vee \neg sprinkler\}$
    - c.f. equivalent to $\{rainy \rightarrow wet, sprinkler \rightarrow wet, \neg rainy \vee \neg sprinkler\}$
- Output: *world* that satisfies all input clauses
  - *world*: truth assignment to propositions
  - e.g., $\{rainy=\mathrm{T}, wet=\mathrm{T}, sprinkler=\mathrm{F}\}$
    - c.f., (*) $\{rainy=\mathrm{T}, wet=\mathrm{T}, sprinkler=\mathrm{T}\}$

Excerpt from http://homes.cs.washington.edu/~pedrod/803/

Markov Logic: Intuition

- A logical KB is a set of **hard constraints** on the set of possible worlds
- Let's make them **soft constraints**: When a world violates a formula, It becomes less probable, not impossible
- Give each formula a **weight** (Higher weight ⇒ Stronger constraint)

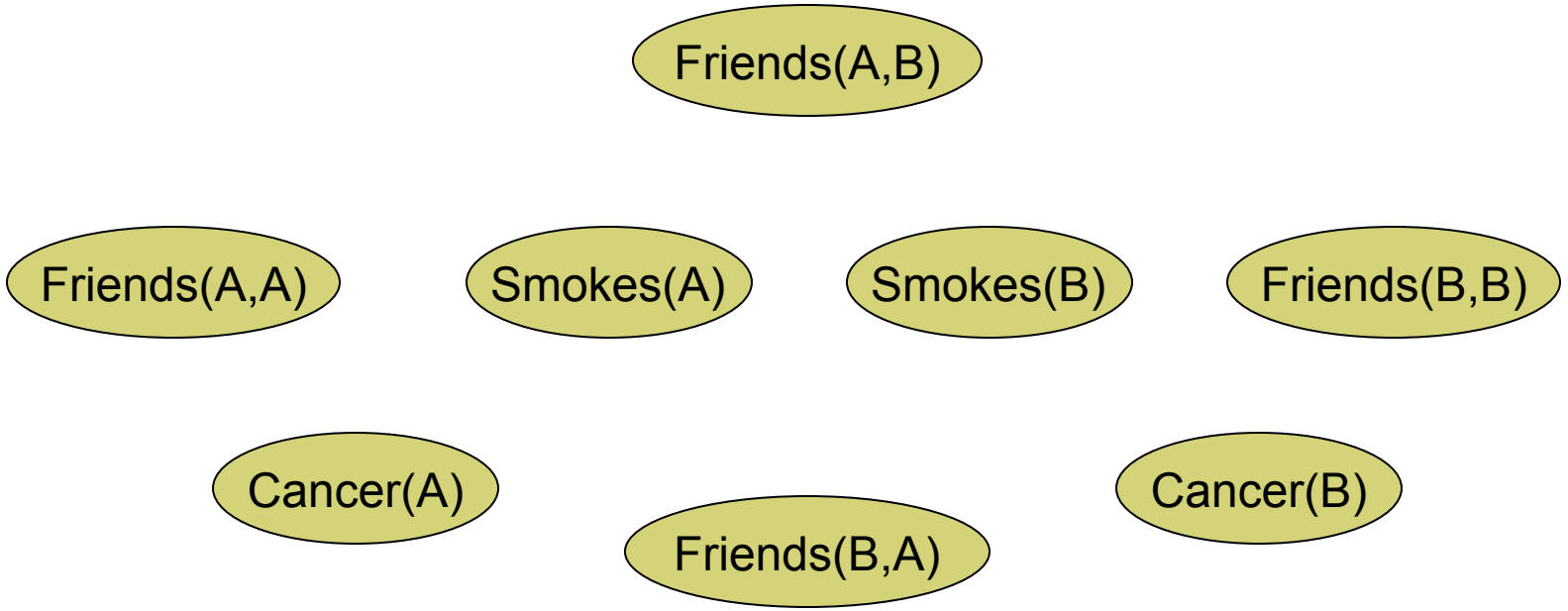
$$P(\text{world}) \propto \exp\left(\sum \text{weights of formulas it satisfies}\right)$$

Excerpt from http://homes.cs.washington.edu/~pedrod/803/

Example: Friends & Smokers

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

Two constants: **Anna** (A) and **Bob** (B)



Excerpt from http://homes.cs.washington.edu/~pedrod/803/

Markov Logic Networks

- MLN is **template** for ground Markov nets
- Probability of a world x:

$$P(x) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(x) \right)$$

Weight of formula *i*

No. of true groundings of formula *i* in *x*

- **Typed** variables and constants greatly reduce size of ground Markov net
- Functions, existential quantifiers, etc.
- Infinite and continuous domains

Excerpt from http://homes.cs.washington.edu/~pedrod/803/

MAP Inference

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

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

Query

Evidence

Excerpt from http://homes.cs.washington.edu/~pedrod/803/

MAP Inference

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

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

- This is just the weighted MaxSAT problem
- Use weighted SAT solver (e.g., MaxWalkSAT [Kautz et al., 1997]

Previous approaches (p. 2)

1. *Ground* first-order knowledge base

x_i in f with X_i. A *ground formula* is a formula obtained by substituting all of its variables with a constant. A *ground KB* is a KB containing all possible groundings of all of its formulas. For example, the grounding of a KB containing one formula, $\texttt{Smokes}(x) \Rightarrow \texttt{Asthma}(x)$ where $\Delta x = \{Ana, Bob\}$, is a KB containing two formulas: $\texttt{Smokes}(Ana) \Rightarrow \texttt{Asthma}(Ana)$ and $\texttt{Smokes}(Bob) \Rightarrow \texttt{Asthma}(Bob)$.

2. Us… so…T

Problem:

Grounding produces a huge search space!

Thus, instead of performing a search for the MAP solution over $O(2^{\sum_{i=1}^{n} d_i})$ assignments as the ground inference algorithms do, we can perform the search over an exponentially smaller $O(\prod_{i=1}^{n}(d_i + 1))$ space, yielding a lifted MAP inference algorithm.

– …5]

– …

– …

– …10]

Solution 1/2: intuition (p.4)

that they contain several worlds with the same probability. We can group together these equi-probable worlds and perform MAP inference by just iterating over the groups, selecting the group with the maximum probability. The fol-

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)

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

equi-probable groups for these assignments. It turns out that each group can be represented by a pair (i, j) where i and j are the number of true groundings of $\mathtt{R}$ and $\mathtt{S}$ respectively. Namely, $i, j \in \{0, 1, 2\}$. Thus, to compute the MAP tuple, we only have to iterate over 9 groups while the ground (naive) MAP inference algorithm will iterate over 16 assignments. In general, the number of groups will be

The above is true *if an MLN is a non-shared*.

Definition 1. *A normal MLN is called* **a non-shared MLN** *if each of its formulas is non-shared. A formula f_i is non-shared if every logical variable appears at most once in the formula. In other words, in a non-shared MLN, no logical variable is shared between the atoms in a formula.*

For example, $\mathtt{R}(x) \vee \mathtt{S}(y)$ is a non-shared formula. However, $\mathtt{R}(x) \vee \mathtt{S}(x)$ is not because x is shared.

Solution 1/2: formal (p.4)

Theorem 1. *Given a non-shared MLN $\mathcal{M}$, let ω_1 and ω_2 be two worlds such that for each atom R in the MLN, the number of true groundings of R in ω_1 is equal to the number of true groundings of R in ω_2. Then,* $\Pr_{\mathcal{M}}(\omega_1) = \Pr_{\mathcal{M}}(\omega_2)$.

(**Proof**: omitted)

Theorem 1 yields the following lifted inference algorithm. Let $\{\mathtt{R}_1, \mathtt{R}_2, \ldots, \mathtt{R}_n\}$ be the atoms in the non-shared MLN. Let d_i denote the domain size of $\mathtt{R}_i$ (the domain of an atom equals the cartesian product of the domains of its logical variables). By Theorem 1, all the ground assignments of the MLN can be grouped into assignments of the form $\langle(\mathtt{R}_i, a_i)|i \in \{1, \ldots, n\}\rangle$ where $a_i \in \{0, \ldots, d_i\}$ and the assignment indicates a_i groundings of $\mathtt{R}_i$ are true. We will refer to $(\mathtt{R}_i, a_i)$ as a **counting assignment** [14]. The algorithm iterates over all tuples of the form: $\langle(\mathtt{R}_1, a_1), \ldots, (\mathtt{R}_n, a_n)\rangle$, computes the weight of the tuple, and returns the tuple with the maximum weight as the MAP tuple. This lifted algorithm is clearly more efficient than its propositional counterpart. The search space over which the propositional algorithm operates is bounded by $O(2^{\sum_{i=1}^{n} d_i})$ where n is the number of atoms in the MLN. On the other hand, the search space of the lifted algorithm is bounded by $O(\prod_{i=1}^{n}(d_i + 1))$. Since the search space is bounded polynomially by the domain size of the logical variables, we have:

前ページの一般化

各アトム R_i について、何個の groundings が true であるかを考えれば十分なので...

Solution 2/2: introduction (p.4)

propositional MAP algorithms, it turns out that we can further reduce the search space for a sub-class of non-shared MLNs, namely non-shared MLNs without self-joins.[2] We

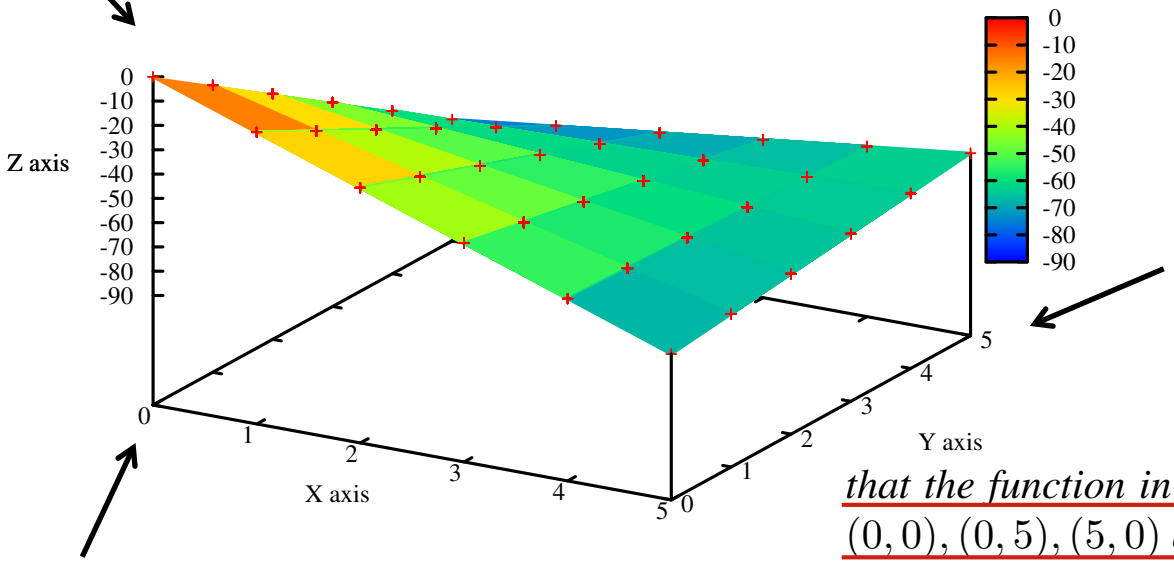
[2] We say that a formula has no self-joins if a predicate symbol appears at most once in the formula.

○: $R(x) \vee S(y)$
×: $R(x) \vee S(y) \vee R(y)$

Solution 2/2: intuition (p.5)

Example 2. *Consider the MLN used in Example 1. Let* $w_1 = -4, w_2 = 5$ *and* $w_3 = 3$. *Assume that the domain of* x, y *is* $\{A, B, C, D, E\}$. *If we iterate through all possible counting assignments to* R *and* S *and plot the total weight of satisfied clauses as a function of counting assignment of* R *and* S *we get the plot in Figure 2. Figure 2 shows*

MAP solution



counting assignment:
<(R, 0), (S, 0)> に対応

counting assignment:
<(R, 0), (S, 5)> に対応

that the function in the plot has only four extreme points: $(0, 0), (0, 5), (5, 0)$ *and* $(5, 5)$. *These extreme points correspond to all groundings of* R *and* S *as either being all true or all false. Since the MAP value can only lie on these extreme points, we only have to evaluate these extreme points for computing the MAP value. It turns out that the MAP tuple is* $\langle (R, 0), (S, 0) \rangle$.

Solution 2/2: formal (p.5)

same truth value. We will refer to this kind of assignment (i.e., all ground atoms having the same truth value) as a **uniform assignment** [1]. This observation, that the atoms

Theorem 3. *For a non-shared MLN without self-joins, in at least one of the MAP solutions, all predicates have uniform assignments.*

(**Proof**: omitted)

Corollary 1. *The MAP inference in a non-shared MLN $\mathcal{M}$ that contains no self-joins can be converted to an equivalent propositional weighted Max-SAT problem with number of variables equal to the number of first order atoms in $\mathcal{M}$.*

Proof. Given a non-shared MLN, $\mathcal{M}$ with m weighted clauses $\{(C_i; w_i)\}_{i=1}^{m}$ that contains no self-joins, we first construct a weighted propositional knowledge base $\mathcal{S}$. We create $\mathcal{S} = \{(C_i'; w_i')\}_{i=1}^{m}$ with $\mathbf{v}$ propositional variables where every $v_k \in \mathbf{v}$ corresponds to a distinct atom $\mathtt{R}_{v_k}$ in $\mathcal{M}$. All atoms in $\mathcal{M}$ have a corresponding variable in $\mathbf{v}$ and vice versa. The assignment $true$ to variable v_k corresponds to the positive uniform assignment to $\mathtt{R}_{v_k}$, i.e. - $(\mathtt{R}_{v_k}, \Delta_{\mathtt{R}_{v_k}})$ and assigning $false$ to variable v_k corresponds to the negative uniform assignment to $\mathtt{R}_{v_k}$, i.e. - $(\mathtt{R}_{v_k}, 0)$. C_i' is constructed by replacing each atom in C_i by its corresponding variable in $\mathbf{v}$. The weight of the clause is computed as $w_i' = \Delta_{C_i} \times w_i$, where Δ_{C_i} is the number of possible groundings of C_i. For uniform assignment, all

1. それぞれの atom に、命題変数 v_k を対応させる。
   v_k=*T*: R_{vk} は positive uniform assignment
   v_k=*F*: R_{vk} は negative uniform assignment
   (...つまり、(R_{vk}, Δ_{Rvk}) or (R_{vk}, 0))

2. C_i のそれぞれのアトムを、R_{vk} で置き換える。
   重みは、$w_i \times \Delta C_i$

3. Weighted MaxSAT solver に投げ v_k の割り当てを求め、対応する atom の真偽値に変換

Includes excerpt from http://en.wikipedia.org/wiki/Unit_propagation

Possible extensions (p.6)

- Unit propagation

$$\{ \quad a \vee b, \qquad \neg a \vee c, \qquad \neg c \vee d, \qquad a \quad \}$$

(removed) ($\neg a$ deleted) (unchanged) (unchanged)

$$\{ \qquad\qquad c, \qquad \neg c \vee d, \qquad a \quad \}$$

- Pure literal elimination

lifted to MAP inference for MLNs, removes (i) Clauses guaranteed to be satisfied for all groundings; and (ii) Atoms guaranteed to be false for all groundings. The following

Proposition 1. *Given an MLN $\mathcal{M}$, if a predicate S appears in k clauses $\mathbf{C} = \{C_i; w_i\}_{i=1}^{k}$, (i) if $w_i \geq 0$, $\forall\ 1 \leq i \leq k$ and S either always occurs as a positive literal or always occurs as a negative literal in $\mathcal{M}$, every $C_i \in \mathbf{C}$ can be removed from $\mathcal{M}$; and (ii) if $w_i < 0$, $\forall\ 1 < i \leq k$ and S either always occurs as a positive literal or always occurs as a negative literal in $\mathcal{M}$, then every occurrence of S can be removed from $\mathcal{M}$.*

How much scalable is it?

For our experiments, we implemented two lifted MAP algorithms, (i) An anytime exact solver based on Integer Linear Programming (L-ILP); and (ii) An anytime approximate solver based on WalkSAT architecture (L-MWS).

We compared both our algorithms with MaxWalkSAT which is the MAP inference algorithm implemented within two state-of-the-art MLN systems, Alchemy (MWS) and Tuffy (TUFFY)[15]. Since both these systems produce approximate results, we implemented an exact MAP inference algorithm using Gurobi (ILP). All three algorithms, MWS, TUFFY and ILP work on the propositional search space, i.e. they ground the entire MLN before performing MAP inference.

Lifted family:
L-ILP
L-MWS

v.s.

Propositional family:
MWS
TUFFY
ILP

Dataset

(i) A **Student** MLN having four formulas:

$\texttt{Teaches}(\textit{teacher,course}) \wedge \texttt{Takes}(\textit{student,course}) \rightarrow \texttt{JobOffers}(\textit{student,company})$; ← !!!?

$\texttt{Teaches}(\textit{teacher,course})$;

$\texttt{Takes}(\textit{student,course})$; and

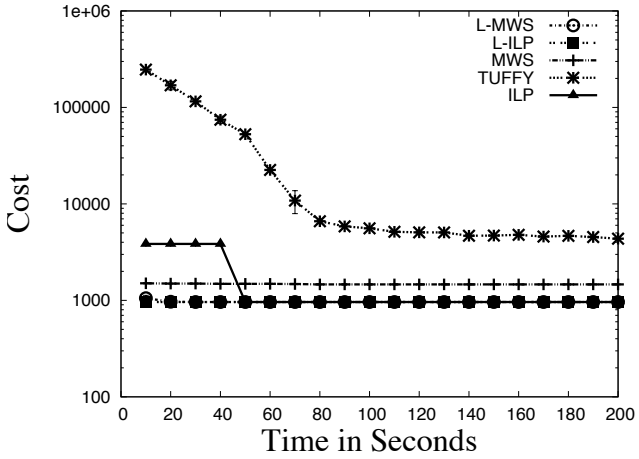
$\neg\texttt{JobOffers}(\textit{student,company})$.

(ii) **WebKB** MLN [12] from the Alchemy web page, consisting of three predicates and six formulas.

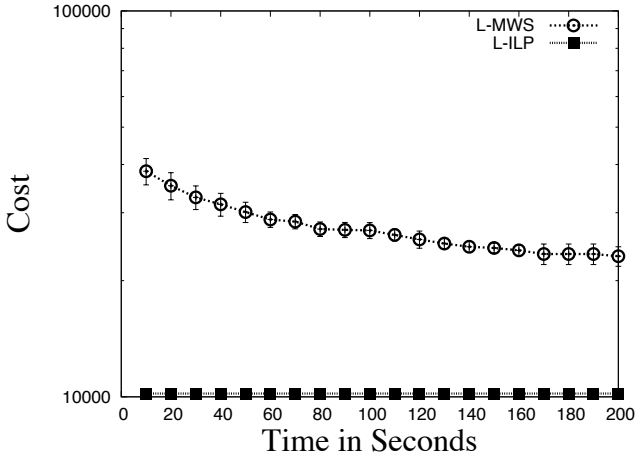
(iii) **Citation Information-Extraction** (IE) MLN [12] from the Alchemy web page, consisting of five predicates and fourteen formulas.

Available at http://alchemy.cs.washington.edu/data/

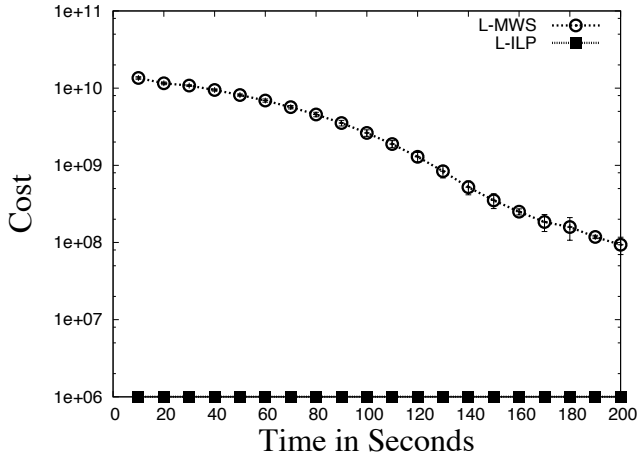
Results



(a) Student-30 (810K clauses)



(b) Student-100 (100 million clauses)



(c) Student-500 (62 Billion clauses)

Cost vs Time: Cost of unsatisfied clauses (smaller is better) against time for benchmark MLNs for different domain sizes.

- For (b), (c), propositional family ran out of memory.
- ILP-based solver is efficient

L-ILP being the superior approach. However, the main virtue of our approach is that *we could use any off-the-shelf solver that is purely propositional in nature to perform lifted inference*. This allows us to scale to large domain-sizes without implementing a new lifted solver. We believe that this abstraction greatly simplifies the development of lifted algorithms by benefitting from the advances made in propositional algorithms.