Lifted MAP Inference for Markov Logic Networks

Somdeb Sarkhel, Deepak Venugopal
Department of Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Parag Singla
Department of CSE
I.I.T. Delhi
New Delhi, India

Vibhav Gogate
Department of Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

In this paper, we present a new approach for lifted MAP inference in Markov Logic Networks (MLNs). Our approach is based on the following key result that we prove in the paper: if an MLN has no shared terms then MAP inference over it can be reduced to MAP inference over a Markov network having the following properties: (i) the number of random variables in the Markov network is equal to the number of first-order atoms in the MLN; and (ii) the domain size of each variable in the Markov network is equal to the number of groundings of the corresponding first-order atom. We show that inference over this Markov network is exponentially more efficient than ground inference, namely inference over the Markov network obtained by grounding all first-order atoms in the MLN. We improve this result further by showing that if non-shared MLNs contain no self joins, namely every atom appears at most once in each of its formulas, then all variables in the corresponding Markov network need only be bi-valued.

Our approach is quite general and can be easily applied to an arbitrary MLN by simply grounding all of its shared terms. The key feature of our approach is that because we reduce lifted inference to propositional inference, we can use any propositional MAP inference algorithm for performing lifted MAP inference. Within our approach, we experimented with two propositional MAP inference algorithms: Gurobi and MaxWalkSAT. Our experiments on several benchmark MLNs clearly demonstrate our approach is superior to ground MAP inference in terms of scalability and solution quality.

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33.

1 Introduction

Statistical relational models [6] such as Markov logic networks [5] bring the power and compactness of first-order logic to probabilistic graphical models. They are routinely used to solve hard problems in a wide variety of real-world application domains including computer vision, natural language processing, robotics and the Web. Recently, there has been a growing interest in exploiting relational structure during inference in statistical relational models. Unlike propositional inference algorithms which operate on individual random variables, these algorithms, which are often called lifted or first-order inference algorithms [17], perform inference over a group of random variables. As a result, they are more scalable, typically exponentially faster when relational structure is present, than propositional inference algorithms.

In this paper, we present a general approach for lifted MAP inference in MLNs. Our approach is based on our key result that when the MLN contains no formulas having shared terms (we refer to such MLNs as a *non-shared* MLN), MAP inference is *domain liftable* [3], namely it is polynomial in the domain size of the logical variables in the MLN. In particular, we show that in non-shared MLNs, the set of full assignments having the cardinality $O(2^{\sum_{i=1}^{n} d_i})$, where n is the number of predicates in the MLN and d_i is the number of possible groundings of the i-th atom in the MLN,[1] can be partitioned into $O(\prod_{i=1}^{n}(d_i+1))$ subsets such that each element in a subset has the same probability. 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.

We further extend this result by showing that if the non-shared MLN has no self joins, namely each atom appears at most once in each formula, then one of the MAP solutions is guaranteed to lie at the following extreme points: for each atom, all of its groundings are either all true or

[1] Number of possible groundings of an atom is the product of the domain sizes of the logical variables appearing in it.

all false. This helps us to further reduce the complexity of inference from $O(\prod_{i=1}^{n}(d_i+1))$ to $O(2^n)$, i.e., the complexity of inference is independent of the domain size.

We utilize the aforementioned results by developing a mapping from the lifted search space to propositional search space of the same size. This helps us reformulate the MAP inference task over non-shared MLNs as propositional MAP inference over a Markov network such that: (i) the number of random variables in the Markov network is equal to the number of atoms in the MLN (i.e., equal to n) and (ii) the domain size of each random variable is either d_i or 2 depending upon whether the non-shared MLN has self-joins or not. The two key features of this formulation are: (i) we can plug in any known propositional MAP inference algorithm for inference on this Markov network; and (ii) since the propositional algorithm operates in the lifted search space, it has the same performance guarantees as a lifted algorithm. Thus, by plugging in different propositional MAP inference algorithms, our approach yields a family of lifted MAP inference algorithms.

Our approach is quite general and can be extended to arbitrary MLNs that have shared terms in a straight-forward way: simply ground all the shared terms of the MLN to obtain an equivalent non-shared MLN. The only caveat is that if all terms in all atoms of the MLN are shared, our approach will have the same complexity as ground inference but never worse (e.g., in the transitive formula $\forall\ x, y, z\ \mathtt{R}(x,y) \wedge \mathtt{R}(y,z) \Rightarrow \mathtt{R}(x,z)$, all terms of R are shared). Note that this is a limitation of not only our approach but lifted inference in general: it is only useful when symmetries are present.

We experimentally evaluated our approach on three benchmark MLNs: WebKB and Information Extraction MLNs available from the Alchemy [12] web page and the Student MLN created by us. We used two state-of-the-art MAP inference algorithms within our approach: (i) Gurobi [8], which is an integer linear programming solver and (ii) MaxWalkSAT [11] which is a popular local search solver. Our experiments clearly show that our approach is significantly better in terms of solution quality and scalability than ground inference. In particular, as we increase the number of objects in the MLN, our algorithms are an order of magnitude faster and better in terms of solution quality.

The rest of the paper is organized as follows. In section 2, we present preliminaries. In section 3, we present our new approach and in section 4, we extend it with several heuristics and pruning techniques. In section 5, we present experimental results and conclude in section 6.

2 Notation and Preliminaries

In this section, we describe notation and preliminaries on propositional logic, first-order logic, Markov logic and MAP inference. For more details, refer to [5, 9, 13].

2.1 Propositional and First-order Logic

The language of propositional logic consists of atomic sentences called propositions or atoms, and logical connectives such as $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (negation), $\Rightarrow$ (implication) and $\Leftrightarrow$ (equivalence). Each proposition takes values from the binary domain {`False, True`} (or $\{0, 1\}$). A propositional formula f is an atom, or any complex formula that can be constructed from atoms using logical connectives. For example, A, B and C are propositional atoms and $f = A \vee \neg B \wedge C$ is a propositional formula. A *knowledge base* (KB) is a set of formulas. A *world* is a truth assignment to all atoms in the KB.

First-order logic (FOL) generalizes propositional logic by allowing atoms to have internal structure; an atom in FOL is a predicate that represents relations between objects. A predicate consists of a predicate symbol, denoted by Monospace fonts (e.g., `Friends`, `Smokes`, etc.), followed by a parenthesized list of arguments called *terms*. A term is a logical variable, denoted by lower case letters (e.g., x, y, etc.), or a constant, denoted by upper case letters (e.g., X, Y, etc.). We assume that each logical variable, say x, is typed and takes values over a finite set (called *domain*) Δx. The language of FOL also includes two quantifiers: $\forall$ (universal) and $\exists$ (existential) which express properties of an entire collection of objects. A formula in first order logic is a predicate (atom), or any complex sentence that can be constructed from atoms using logical connectives and quantifiers. For example, the formula $\forall x$ $\mathtt{Smokes}(x) \Rightarrow \mathtt{Asthma}(x)$ states that all persons who smoke have asthma. Where as $\exists x$ $\mathtt{Cancer}(x)$ states that there exists a person x who has cancer.

In this paper, we use a subset of FOL which has no function symbols, equality constraints or existential quantifiers. We also assume that domains are finite (and therefore function-free) and that there is a one-to-one mapping between constants and objects in the domain (Herbrand interpretations). We assume that each formula f is of the form $\forall$x f, where x are the set of variables in f and f is a conjunction or disjunction of literals; each literal being an atom or its negation. For brevity, we will drop $\forall$ from all the formulas. Given variables $\mathbf{x} = \{x_1, ..., x_n\}$ and constants $\mathbf{X} = \{X_1, ..., X_n\}$ where $X_i \in \Delta x_i$, $f[\mathbf{X}/\mathbf{x}]$ is obtained by substituting every occurrence of variable 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, $\mathtt{Smokes}(x) \Rightarrow \mathtt{Asthma}(x)$ where $\Delta x = \{Ana, Bob\}$, is a KB containing two formulas: $\mathtt{Smokes}(Ana) \Rightarrow \mathtt{Asthma}(Ana)$ and $\mathtt{Smokes}(Bob) \Rightarrow \mathtt{Asthma}(Bob)$.

2.2 Markov Logic

Markov logic [5] extends FOL by softening the hard constraints expressed by the formulas and is arguably the most popular modeling language for SRL. A soft formula or a weighted formula is a pair (f, w) where f is a formula in FOL and w is a real-number. A Markov logic network (MLN), denoted by $\mathcal{M}$, is a set of weighted formulas (f_i, w_i). Given a set of constants that represent objects in the domain, a Markov logic network defines a Markov network or a log-linear model. The Markov network is obtained by grounding the weighted first-order knowledge base and represents the following probability distribution.

$$P_{\mathcal{M}}(\omega) = \frac{1}{Z(\mathcal{M})} \exp\left(\sum_i w_i N(f_i, \omega)\right) \quad (1)$$

where ω is a world, $N(f_i, \omega)$ is the number of groundings of f_i that evaluate to $\texttt{True}$ in the world ω and $Z(\mathcal{M})$ is a normalization constant or the partition function.

Throughout the paper, we will assume that the input MLN to our algorithm is in normal form [10]. We require this for simplicity of exposition. A *normal* MLN is an MLN that satisfies the following two properties: (1) There are no constants in any formula, and (2) If two distinct atoms with the same predicate symbol have variables x and y in the same position then $\Delta x = \Delta y$. Any MLN can be converted to a normal MLN. Note that in a normal MLN, we assume that the terms in each atom are ordered and therefore we can identify each term by its position in the order. Furthermore, we assume that the MLN is expressed as a set of weighted clauses.

2.3 MAP Inference in MLNs

A common optimization inference task over MLNs is finding the most probable state of the world ω, that is finding a complete assignment to all ground atoms which maximizes the probability. This task is known as *Maximum a Posteriori* (MAP) inference in the Markov network literature, and *Most Probable Explanation* (MPE) inference in the Bayesian network literature. For Markov logic, this is formally defined as follows:

$$\begin{aligned} \arg\max_{\omega} P_{\mathcal{M}}(\omega) &= \arg\max_{\omega} \frac{1}{Z(\mathcal{M})} \exp\left(\sum_i w_i N(f_i, \omega)\right) \\ &= \arg\max_{\omega} \sum_i w_i N(f_i, \omega) \end{aligned} \quad (2)$$

From Eq. (2), we can see that the MAP problem in Markov logic reduces to finding the truth assignment that maximizes the sum of weights of satisfied clauses. Therefore, any weighted satisfiability solver can be used to solve this problem. The problem is NP-hard in general, but effective solvers exist, both exact and approximate. Examples of such solvers are MaxWalkSAT, a weighted variant of the WalkSAT local-search satisfiability solver [18] and Clone [16], a branch-and-bound solver. However, all of these algorithms are propositional and therefore they cannot exploit relational structure that is inherent to MLNs.

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: $[\texttt{R}(x) \vee \texttt{S}(y), w_1]$; $[\texttt{R}(x), w_2]$; and $[\texttt{S}(y), w_3]$ with domains given by $\Delta_x = \Delta_y = \{A, B\}$.

3 Lifted Formulation of MAP Inference

In this section, we show that we can reduce MAP inference in a sub-class of MLNs, which we call *non-shared MLNs*, to MAP inference over an equivalent propositional MLN (or a propositional Markov network) *such that the number of propositional variables in the propositional MLN is equal to the number of first order atoms in the non-shared MLN*. This is in contrast to ground MAP inference in which the number of propositional variables is equal to the number of ground atoms.

We begin by defining non-shared MLNs.

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

3.1 Domain-Lifted MAP Inference over Non-Shared MLNs

We show that MAP inference over non-shared MLNs is *domain liftable* [3], namely inference over it is polynomial in the domain size of the logical variables in the MLN. The

key reason that non-shared MLNs are domain liftable is 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 following example illustrates this grouping.

Example 1. *Consider the non-shared MLN containing three formulas:* $[\mathtt{R}(x) \vee \mathtt{S}(y), w_1]$*;* $[\mathtt{R}(x), w_2]$*; and* $[\mathtt{S}(y), w_3]$*. Let* $\Delta_x = \Delta_y = \{A, B\}$*. Figure 1 gives a truth table showing all possible assignments to the ground atoms as well as their weights. Figure 1 also shows nine 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 equal to* $(|\Delta_x|+1)(|\Delta_y|+1)$ *while the number of possible assignments to the ground atoms equals* $2^{|\Delta_x|+|\Delta_y|}$.

We can generalize the ideas presented in Example 1 using the following theorem:

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

Proof. We will prove this theorem by leveraging the generalized binomial rule [10]. The generalized Binomial rule states that if an atom $\mathtt{R}$ is non-shared (which is a special case of singleton atoms), then the MLNs obtained by conditioning on the following subset of assignments to all groundings of $\mathtt{R}$ are equivalent: the number of true groundings of $\mathtt{R}$ is the same in all the assignments in the subset. Moreover, according to the rule, the following two conditions hold:

- if the MLN is non-shared then the new MLN is also non-shared
- the number of formulas involving $\mathtt{R}$ satisfied by each assignment in the subset is the same.

Let $\mathtt{R}_1, \ldots, \mathtt{R}_n$ be the atoms in the MLN $\mathcal{M}$. Let d_i be the domain size of $\mathtt{R}_i$. Let $\mathtt{R}_i = j_{i,1}$ and $\mathtt{R}_i = j_{i,2}$ where $j_{i,k} \in 2^{d_i}$, $k \in \{1, 2\}$ denote the assignment to all groundings of $\mathtt{R}_i$ in the worlds ω_1 and ω_2 respectively. Let us condition the atoms along the order $\mathtt{R}_1, \ldots, \mathtt{R}_n$. By the generalized Binomial rule, the MLN obtained by conditioning on $\mathtt{R}_1 = j_{1,1}$, denoted by $\mathcal{M}|_{\mathtt{R}_1 = j_{1,1}}$ is equivalent to the MLN $\mathcal{M}|_{\mathtt{R}_1 = j_{1,2}}$ obtained by conditioning on $\mathtt{R}_1 = j_{1,2}$ (since the number of true groundings of $\mathtt{R}_i$ is the same in both the assignments). Let $w(\mathcal{M}|_{\mathtt{R}_i = j_{i,k}})$, $k \in \{1, 2\}$ denote the sum of the weights of clauses satisfied by conditioning on the assignment $\mathtt{R}_i = j_{i,k}$. By the generalized Binomial rule, $w(\mathcal{M}|_{\mathtt{R}_1 = j_{1,1}}) = w(\mathcal{M}|_{\mathtt{R}_1 = j_{1,2}})$. Moreover, since all atoms in $\mathcal{M}|_{\mathtt{R}_1 = j_{1,1}}$ and $\mathcal{M}|_{\mathtt{R}_2 = j_{2,1}}$ are non-shared, it follows that the MLNs obtained by further conditioning on $\mathtt{R}_2 = j_{2,1}$ is the same as the one obtained by conditioning on $\mathtt{R}_2 = j_{2,2}$. By iteratively (inductively) applying this argument, we have:

$$\sum_{i=1}^{n} w(\mathcal{M}|_{\mathtt{R}_1 = j_{1,1}, \ldots, \mathtt{R}_{i-1} = j_{i-1,1}}) = \sum_{i=1}^{n} w(\mathcal{M}|_{\mathtt{R}_1 = j_{1,2}, \ldots, \mathtt{R}_{i-1} = j_{i-1,2}}) \quad (3)$$

In other words, the two worlds ω_1 and ω_2 have the same weight. Therefore, $\Pr_{\mathcal{M}}(\omega_1) = \Pr_{\mathcal{M}}(\omega_2)$. □

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:

Theorem 2. *MAP inference in non-shared MLNs is domain liftable.*

Although this represents a significant improvement over 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 will present this result next.

3.2 MAP Inference over non-shared MLNs without Self-Joins

We illustrate the main idea in our result using the following example.

Example 2. *Consider the MLN used in Example 1. Let* $w_1 = -4, w_2 = 5$ *and* $w_3 = 3$*. Assume that the domain*

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

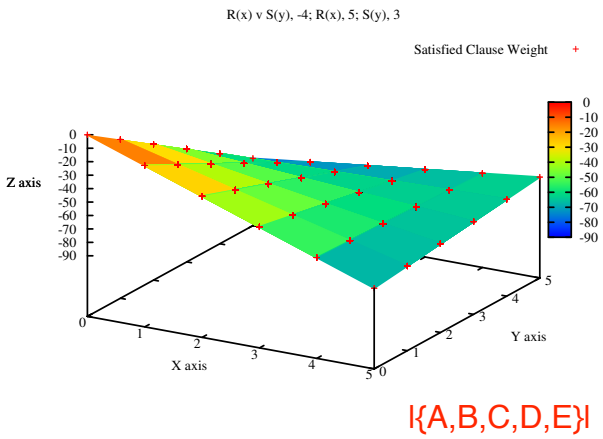


Figure 2: The total weight of satisfied clauses as a function of counting assignment of R and S. The plane is for illustration purpose only.

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 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 $\langle(R, 0), (S, 0)\rangle$.

We observe in the previous example that all the ground atoms of the predicate R (and the predicate S) have the 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 have a uniform assignment in the MAP state, holds not only for this example but for any non-shared MLN without self-joins, and we will prove this formally next.

Lemma 1. *The sum of weights of satisfied clauses for a non-shared MLN without self-join is a multilinear function on the counting assignment of its predicates.*

Proof. (*Sketch*) Consider a non-shared MLN $\mathcal{M}$ that contains m weighted clauses $\{(C_i; w_i)\}_{i=1}^{m}$. Let $\mathbf{V}(C_i)$ represent the set of all the atoms in the clause C_i. Let $\mathbf{V}^{(+)}(C_i)$ represent the set of atoms which appear as positive literals in C_i. Let $\mathbf{V}^{(-)}(C_i)$ represent the set of atoms appearing as negative literals. Given an atom R, let $(\mathrm{R}, v_{\mathrm{R}})$ denote its counting assignment. It can be easily shown that the number of groundings of C_i that are unsatisfied by the counting assignment is given by,

$$\prod_{\mathrm{R}\in\mathbf{V}^{(+)}(C_i)} (\Delta_{\mathrm{R}} - v_{\mathrm{R}}) \prod_{\mathrm{R}\in\mathbf{V}^{(-)}(C_i)} v_{\mathrm{R}}$$

where Δ_{R} represents the number of possible groundings of R. Clearly, the total number of possible groundings of C_i is equal to $\prod_{\mathrm{R}\in C_i}(\Delta_{\mathrm{R}})$. Therefore, the sum of weights of satisfied clauses for $\mathcal{M}$ is given by,

$$\sum_{C_i} w_i \Big(\prod_{\mathrm{R}\in C_i} (\Delta_{\mathrm{R}}) - \prod_{\mathrm{R}\in\mathbf{V}^{(+)}(C_i)} (\Delta_{\mathrm{R}} - v_{\mathrm{R}}) \prod_{\mathrm{R}\in\mathbf{V}^{(-)}(C_i)} v_{\mathrm{R}}\Big) \quad (4)$$

Clearly eq. 4 is a multilinear function in v_R since v_R never appears more than once in the product term (if there are no self-joins in $\mathcal{M}$). □

Lemma 2. *Consider a multilinear function $\Phi(\mathbf{v})$ defined over a tuple of variables $\mathbf{v} = (v_1, v_2, \cdots, v_n)$. Let each v_j take values from the set $\{0, 1, 2, \cdots \Delta_{v_j}\}$. Then, at least one of the solutions $\mathbf{v}^*$ to the optimization problem $\arg\max_{\mathbf{v}} \Phi(\mathbf{v})$ is such that each v_j^* lies at the extremes i.e. $v_j^* = 0$ or $v_j^* = \Delta_j \; \forall j$.*

Proof. We will prove the theorem using induction over n, the number of variables over which the multilinear function is defined. Clearly, the theorem holds true for $n = 1$ since a linear function of one variable has its maxima at the extremes. Assume the theorem holds for any multilinear function defined over $n - 1$ variables. Consider the function $\Phi(\mathbf{v})$ over the variables $\mathbf{v} = (v_1, v_2, \cdots v_n)$. By re-arranging terms, we can write:

$$\max_{\mathbf{v}} \Phi(\mathbf{v}) = \max_{\mathbf{v}\setminus v_n} (\max_{v_n} \Phi(\mathbf{v}))$$

Since $\Phi(\mathbf{v})$ is a multilinear function, it can be seen as a linear function of v_n (holding other variables as constant). Hence, the inner expression on the right side is optimized at an extreme value of v_n ($v_n = 0$ or $v_n = \Delta_{v_n}$). Let $\Phi_0(\mathbf{v} \setminus v_n)$ and $\Phi_{\Delta_{v_n}}(\mathbf{v} \setminus v_n)$, respectively, be the two possible resulting functions by substituting the values of 0 and Δ_{v_n}, for v_n in $\Phi(\mathbf{v})$. In both the cases, we get a new function which is multilinear over $n - 1$ variables. Using the induction hypothesis, its maxima will lie at the extreme values of $v_i, v_2, \cdots v_{n-1}$. Hence, one of the maxima of the original function $\phi(\mathbf{v})$ will lie at the extreme values of $v_1, v_2, \cdots v_n$. Hence, proved. □

Note that above lemma states that at least one of the maxima of a multilinear function will lie at its extremes. It is still possible that there are other maxima which do not lie at the extremes (for instance, think of a constant function). As long as we are interested in finding one of them, above lemma can be put to use. Lemma 1 and Lemma 2 allow us to prove our second main result stated below.

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

We can use Theorem 3 to formulate the MAP problem as a weighted Max-SAT problem as formalized in the following corollary.

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 groundings of each clause C_i is either satisfied or none of them are satisfied. Since whenever C_i is satisfied C_i' is also satisfied and as the weight of C_i' is $\Delta_{C_i} \times w_i$ the sum of weights of satisfied clauses for a complete assignment will be same in both $\mathcal{M}$ and $\mathcal{S}$. As theorem 3 proves that the MAP solution consist of uniform assignments, it follows from equation 2 that the MAP inference in $\mathcal{M}$ is equivalent to solving the weighted Max-SAT problem over $\mathcal{S}$. Hence the corollary follows. □

The result of corollary 1 allows us to use any weighted Max-SAT solver to compute the MAP solution of a non-shared MLN without self-joins. This observation also means that for such MLNs, the optimal solution is independent of the number of objects in the MLN which makes MAP inference especially efficient in these cases.

4 Extensions

In this section, we propose heuristics to make our approach more practical. These heuristics can be considered as pruning techniques, which allow us to greatly reduce the size of the knowledge base. Moreover these heuristics can be applied to any arbitrary MLN. These can give us orders of magnitude of speedup. We propose to use these heuristics as a preprocessing step to simplify the MLN.

4.1 Unit Propagation

Repeated use of unit propagation [4] is one of the key component of highly effective propositional satisfiability testing solvers. The idea in unit propagation is to resolve all clauses with unit clauses, and continue to do this until convergence, i.e., no further unit resolutions are possible. Although this heuristic is very effective for SAT solvers, for Max-SAT, it is not sound. However, this rule can be used for hard unit clauses. We can lift this rule in a straight forward manner, by resolving the hard unit clauses with other clauses. This heuristic in conjunction with the pure literal heuristic can greatly reduce the size of the MLN.

4.2 Pure Literal Elimination

The pure literal elimination rule for SAT formulas [4] when 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 specifies the pure literal elimination rule for MLNs.

Proposition 1. *Given an MLN $\mathcal{M}$, if a predicate $\mathtt{S}$ 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 $\mathtt{S}$ 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 $\mathtt{S}$ either always occurs as a positive literal or always occurs as a negative literal in $\mathcal{M}$, then every occurrence of $\mathtt{S}$ can be removed from $\mathcal{M}$.*

5 Experiments

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 implemented L-ILP using a parallelized ILP solver called Gurobi [8] and implemented L-MWS using MaxWalkSAT [18], a randomized local-search algorithm. 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.

We used three MLNs to evaluate our system,

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

$\mathtt{Teaches}(\textit{teacher,course}) \wedge \mathtt{Takes}(\textit{student,course}) \rightarrow \mathtt{JobOffers}(\textit{student,company})$;

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

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

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

(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.

In order to compare the performance and scalability of our algorithms, we ran two sets of experiments illustrated in Fig. 3 and Fig. 4. Fig. 3, plots the solution quality (total weight of false clauses) achieved by each algorithm for varying time-bounds. Fig. 4 plots the relative-gap between the optimal solution and the solution output by each algorithm, for varying domain-sizes. We describe the results of both our experiments below. All our experiments were run on a quad-core CentOS machine with 8GB RAM.

5.1 Cost vs Time

The results for the Student MLN are shown in Fig. 3 (a)-(c). We see that the lifted algorithms L-ILP and L-MWS are the best performing algorithms for all domain-sizes. At higher domain-sizes (100 and 500), the propositional solvers ILP, MWS and TUFFY ran out of memory. The performance of L-MWS was similar to L-ILP for domain-size equal to 30. For domain-sizes of 100 and 500, L-MWS gradually converges towards the optimal solution, whereas L-ILP was able to exactly solve the problem in less than 10 seconds.

The results for WebKB are shown in Fig. 3 (d)-(f). Again, we can see that the lifted algorithms L-ILP and L-MWS outperform the propositional algorithms and are much more scalable. For the larger domain-sizes (100 and 500), MWS and TUFFY run out of memory. For domain-size 30 and 100, the performance of both the lifted algorithms L-ILP and L-MWS is quite similar.

The results for IE are shown in Fig. 3 (g)-(i). For domain-size 30, the performance of both the lifted algorithms L-ILP and L-MWS is quite similar. For the domain-sizes 100 and 500, L-ILP was able to find the optimal solution while L-MWS was far from optimal.

5.2 Accuracy vs Domain-Size

Fig. 4 illustrates the variation in accuracy for each algorithm as the domain-size increases. Here, we gave each algorithm a fixed time-bound of 500 seconds and measured the relative-gap between the optimal solution (opt) and the best cost given by the algorithm (c) using $\frac{|opt-c|}{opt}$. We see that both L-MWS and L-ILP are quite accurate and scale to much larger domain-sizes. On the other hand, there is a noticeable drop in the accuracy of the propositional algorithms, MWS, TUFFY and ILP as we increase the domain-size. For larger domain-sizes, the propositional algorithms run out of memory.

In summary, our experiments show that our two lifted algorithms L-MWS and L-ILP are far more scalable and accurate than propositional approaches. Since the two approaches are fundamentally different, L-ILP is a complete anytime solver while L-MWS is an approximate solver, as expected they perform differently on the benchmarks, with 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.

6 Summary and Future work

In this paper, we proposed a general approach for lifting MAP inference in Markov Logic Networks (MLNs). We identified cases in which we can reduce lifted MAP inference to inference over an equivalent propositional theory such that the number of propositional variables is equal to the number of first order atoms in the MLN. We used this observation in a straight-forward manner: convert the MLN to an equivalent propositional theory and then apply any propositional algorithm to solve it. For our experiments, we used two propositional algorithms, a complete, anytime algorithm (Gurobi) based on Integer Linear Programming (ILP) and a local-search algorithm called MaxWalksat. Our experiments clearly demonstrate the scalability and promise of our approach.

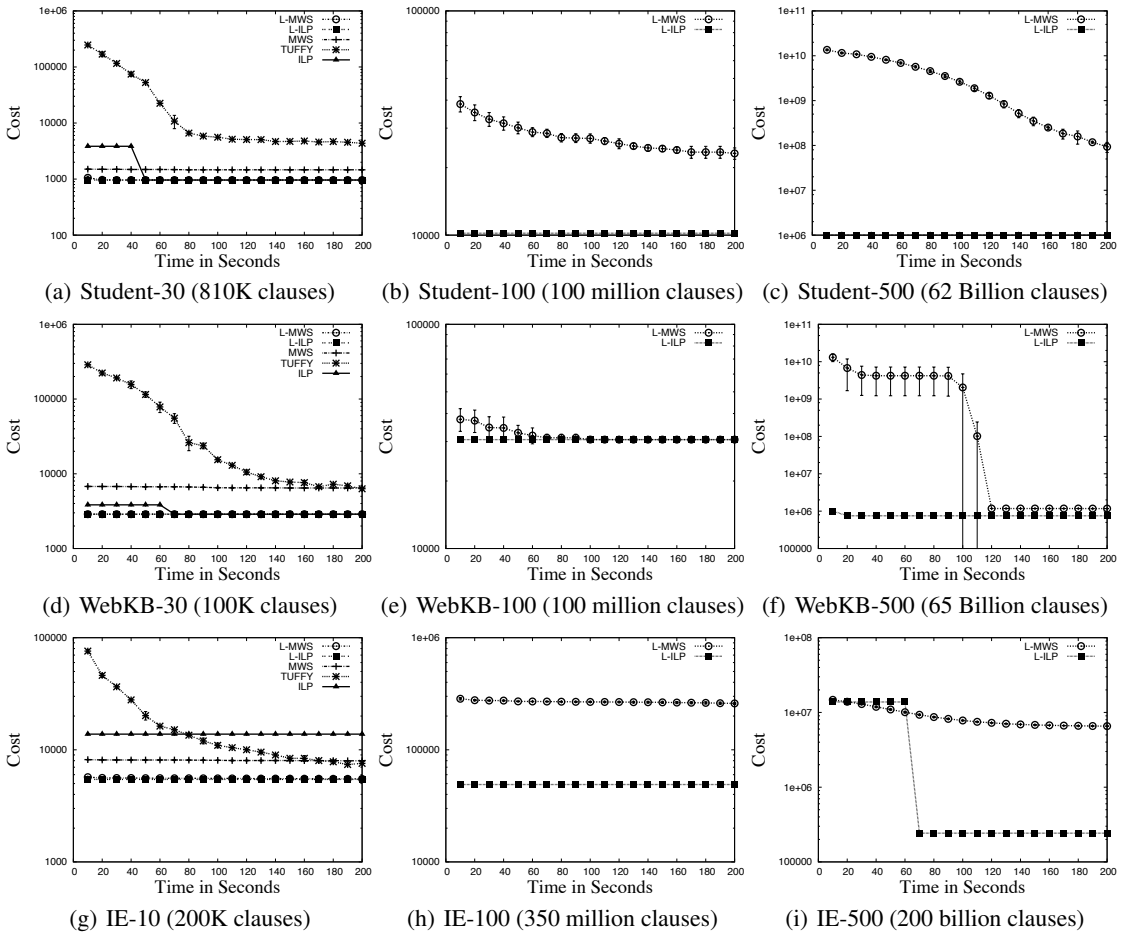
Directions for future work include: combining our approach with other lifted inference rules such as the power rule [7, 10]; identifying cases where our generalized theorem can be applied; applying the results in this paper to lifted MCMC approaches [19]; and using our approach for exploiting symmetries in probabilistic graphical models.

Acknowledgements

This research was partly funded by ARO MURI grant W911NF-08-1-0242, by the AFRL under contract number FA8750-14-C-0021 and by the DARPA Probabilistic Programming for Advanced Machine Learning Program under AFRL prime contract number FA8750-14-C-0005. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of DARPA, AFRL, ARO or the US government.

References

[1] Apsel, U.; and Brafman, R. 2012. Exploiting Uniform Assignments in First-Order MPE. In *Proceedings of the*

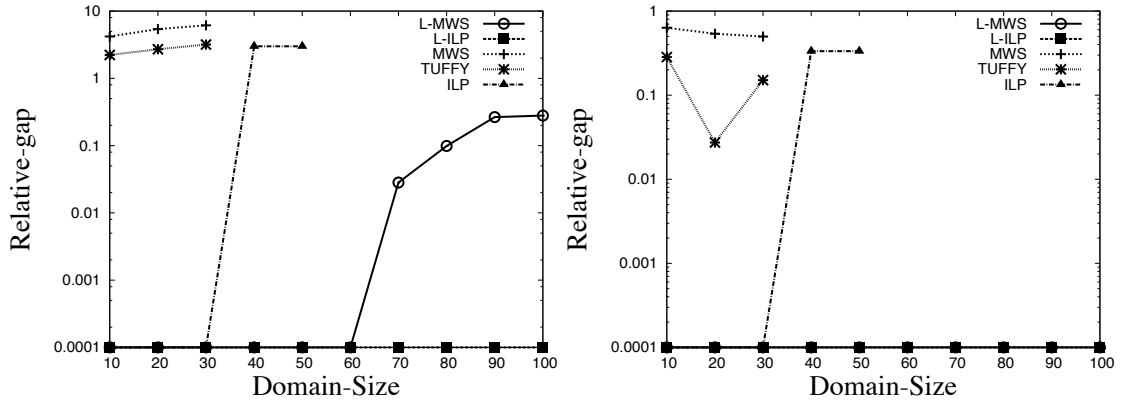


(a) Student-30 (810K clauses) (b) Student-100 (100 million clauses) (c) Student-500 (62 Billion clauses)

(d) WebKB-30 (100K clauses) (e) WebKB-100 (100 million clauses) (f) WebKB-500 (65 Billion clauses)

(g) IE-10 (200K clauses) (h) IE-100 (350 million clauses) (i) IE-500 (200 billion clauses)

Figure 3: Cost vs Time: Cost of unsatisfied clauses (smaller is better) against time for benchmark MLNs for different domain sizes. Notation used to label each figure: MLN-domainsize(number of ground clauses in the MLN). The standard deviation is plotted as error bars. For (b),(c),(e),(f),(h) and (i), no results could be obtained for propositional algorithms since they ran out of memory.



(a) Student (b) WebKB

Figure 4: Accuracy vs Domain-Size: The relative-gap i.e. $\frac{|optimal-cost|}{optimal}$ is plotted for varying domain-sizes (smaller is better). Every algorithm was given 500 seconds of running-time. The propositional algorithms TUFFY, MWS and ILP run out of memory for larger domain-sizes. **Note**: We could not run this experiment for IE since the propositional algorithms ran out of memory when the domain-size exceeded 10.

Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, 74–83. AUAI Press.

[2] Braz, R.,; Amir, E; and Roth, D. 2005. Lifted first-order probabilistic inference. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence* 1319-1325.

[3] Van den Broeck, G., 2011. On the completeness of first-order knowledge compilation for lifted probabilistic inference. In *Advances in Neural Information Processing Systems 24*, 1386–1394.

[4] Davis, M., and Putnam, H. 1960. A computing procedure for quantification theory. In *Journal of the ACM (JACM), 7(3)* 201-215.

[5] Domingos, P., and Lowd, D. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. San Rafael, CA: Morgan & Claypool.

[6] Getoor, L., and Taskar, B., eds. 2007. *Introduction to Statistical Relational Learning*. MIT Press.

[7] Gogate, V., and Domingos, P. 2011. Probabilistic Theorem Proving. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, 256–265. AUAI Press.

[8] Gurobi Optimization, Inc. 2013. Gurobi Optimizer Reference Manual. *http://www.gurobi.com*

[9] Hoos, H. H., and Stützle, T. 2004. *Stochastic local search: Foundations & applications*. Morgan Kaufmann.

[10] Jha, A.; Gogate, V.; Meliou, A.; and Suciu, D. 2010. Lifted Inference from the Other Side: The tractable Features. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS)*, 973–981.

[11] Kautz, H.; Selman, B.; and Jiang, Y. 1997. A General Stochastic Approach to Solving Problems with Hard and Soft Constraints. In Gu, D.; Du, J.; and Pardalos, P., eds., *The Satisfiability Problem: Theory and Applications*. New York, NY: American Mathematical Society. 573–586.

[12] Kok, S.; Sumner, M.; Richardson, M.; Singla, P.; Poon, H.; and Domingos, P. 2006. The Alchemy System for Statistical Relational AI. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA. http://alchemy.cs.washington.edu.

[13] Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

[14] Milch, B.; Zettlemoyer, L. S.; Kersting, K.; Haimes, M.; and Kaelbling, L. P. 2008. Lifted Probabilistic Inference with Counting Formulas. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence* 1062-1068.

[15] Niu, F.; Re , C.; Doan, A.; and Shavlik, J. 2011. Tuffy: Scaling up statistical inference in markov logic networks using an rdbms. In *Proceedings of the VLDB Endowment 4(6)* 373 384.

[16] Pipatsrisawat, K.; and Darwiche, A. 2007. Clone: Solving Weighted Max-SAT in a Reduced Search Space. In *Proceedings of 20th Australian Joint Conference on Artificial Intelligence (AI07)* 223–233.

[17] Poole, D. 2003. First-Order Probabilistic Inference. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 985–991. Acapulco, Mexico: Morgan Kaufmann.

[18] Selman, B.; Kautz, H.; and Cohen, B. 1996. Local Search Strategies for Satisfiability Testing. In Johnson, D. S., and Trick, M. A., eds., *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*. Washington, DC: American Mathematical Society. 521–532.

[19] Venugopal, D.; and Gogate, V. 2012. On Lifting the Gibbs Sampling Algorithm. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)* 1664-1672.