Lifted Hinge-Loss Markov Random Fields

Sriram Srinivasan
ssriniv9@ucsc.edu
UC Santa Cruz

Behrouz Babaki
Behrouz.Babaki@polymtl.ca
Polytechnique Montreal

Golnoosh Farnadi
gfarnadi@ucsc.edu
UC Santa Cruz

Lise Getoor
getoor@ucsc.edu
UC Santa Cruz

Abstract

Statistical relational learning models are powerful tools that combine ideas from first-order logic with probabilistic graphical models to represent complex dependencies. Despite their success in encoding large problems with a compact set of weighted rules, performing inference over these models is often challenging. In this paper, we show how to effectively combine two powerful ideas for scaling inference for large graphical models. The first idea, lifted inference, is a well-studied approach to speeding up inference in graphical models by exploiting symmetries in the underlying problem. The second idea is to frame Maximum a posteriori (MAP) inference as a convex optimization problem and use alternating direction method of multipliers (ADMM) to solve the problem in parallel. A well-studied relaxation to the combinatorial optimization problem defined for logical Markov random fields gives rise to a *hinge-loss Markov random field* (HL-MRF) for which MAP inference is a convex optimization problem. We show how the formalism introduced for coloring weighted bipartite graphs using a color refinement algorithm can be integrated with the ADMM optimization technique to take advantage of the sparse dependency structures of HL-MRFs. Our proposed approach, *lifted hinge-loss Markov random fields* (LHL-MRFs), preserves the structure of the original problem after lifting and solves lifted inference as distributed convex optimization with ADMM. In our empirical evaluation on real-world problems, we observe up to a three times speed up in inference over HL-MRFs.

1 Introduction

Statistical relational learning (SRL) frameworks compactly specify a probability distribution over groups of objects using first-order logic. Most commonly, the probability distribution is defined as a template for a graphical model which is instantiated (or *grounded*) over the objects in the domain. A variety of different SRL frameworks have been developed over the last decade, (see e.g., (De Raedt and Kersting 2011; Richardson and Domingos 2006; Getoor and Taskar 2007)). In this paper, we focus on hinge-loss Markov Random fields (HL-MRFs) (Bach et al. 2017), a recently introduced SRL framework based on weighted logical rules which makes inference tractable by defining a convex inference objective. HL-MRF have been used successfully in a wide variety of domains including NLP tasks (Beltagy, Erk, and Mooney 2014; Wang and Ku 2016), image processing (Aditya, Yang, and Baral 2018; Gridach, Haddad, and Mulki 2017), bioinformatics (Sridhar, Fakhraei, and Getoor 2016), search (Alshukaili, Fernandes, and Paton 2016), recommender systems (Kouki et al. 2017; Lalithsena et al. 2017) and more (Deng and Wiebe 2015; Ebrahimi, Dou, and Lowd 2016; Chen, Chen, and Qian 2014), with promising results.

For SRL frameworks, exact inference is often computationally expensive because inference is performed over large grounded graphical models. However, this ground representation is typically derived from a much smaller set of logical rules and, depending on the data, often contains identical substructures. These identical substructures cause unnecessary work for the inference algorithm by repeatedly performing the same operations.

Lifted inference (Kersting 2012; Kimmig, Mihalkova, and Getoor 2015; den Broeck et al. 2011; Kazemi and Poole 2016) aims to detect common substructures and uses them to avoid redundant computations.

Lifted inference in SRL is a well-studied problem. A popular approach is to group objects that are indistinguishable given evidence, and perform inference by operating on these groups.

First-order variable elimination (Poole 2003; de Salvo Braz, Amir, and Roth 2005) extends the standard variable elimination algorithm by summing over entire groups of random variables instead of one at a time. Lifted belief propagation (Singla and Domingos 2008) employs the same message-passing method as the standard belief propagation algorithm. It first groups variables and forms super nodes which are connected via so-called super edges. Message passing is then performed over the graph with these super nodes and super edges. A modified version of belief propagation (BP) called counting BP (Kersting, Ahmadi, and Natarajan 2009) constructs a compressed factor graph by creating clusternodes and clusterfactors and uses a modified BP to perform inference. Some inference algorithms use the logical structure in a model for problem decomposition (Dechter and Mateescu 2007; Chavira and Darwiche 2008). The lifted versions of these algorithms perform this decomposition at the first-order level (Gogate and Domingos 2010;

den Broeck et al. 2011).

The *exact* lifting methods discussed above assume that variables in the problem of interest are discrete. This makes them inapplicable to languages such as PSL, which are defined over continuous random variables. Recently developed lifted linear programming (Mladenov, Ahmadi, and Kersting 2012; Mladenov, Kersting, and Globerson 2014) and lifted convex quadratic programming (Mladenov, Kleinhans, and Kersting 2017) offer a method for finding and exploiting symmetries in linear programming and quadratic problems. Lifted linear and quadratic programming groups indistinguishable variables using the color refinement algorithm to produce a smaller linear or quadratic program making inference faster.

The inference algorithm in HL-MRFs relies on *alternating direction method of multipliers (ADMM)*. ADMM is an iterative optimization method (Boyd et al. 2011) that provides an elegant approach for finding the saddle point in augmented Lagrangian. The ADMM algorithm for HL-MRFs use the structure in the objective function and solves the subproblems in each iteration using closed-form solutions.

Our work integrates the concept of lifting using the color refinement algorithm with ADMM to perform a more efficient inference in HL-MRFs. Using ADMM for HL-MRFs (Bach et al. 2017) shows exponential performance gains over traditional LP/QP solvers. To our best knowledge this is the first approach that combines ADMM with color refinement to perform lifting for probabilistic inference.

Our contributions are as follows: 1) we propose the first method for detecting and eliminating the symmetries in HL-MRFs inference problems using the color refinement algorithm, By applying this method to the real-world datasets, we observe significant reductions (up to 66%) in the size of problems; 2) we show how the lifted problem can be cast back into the same form as the original inference problem and solved using the specialized inference algorithm of HL-MRFs. The proposed integration of lifted inference and ADMM is essential to our goal. We compare solving the lifted problem with existing off-the-shelf solvers and the ADMM method, and demonstrate that lifting has a better pay off when the latter is employed; 3) we run a series of experiments on synthetically generated data, analyze the complicated relationships graph structures have with lifting, and show the effectiveness of LHL-MRFs on varied levels of symmetry.

2 Background

In this section, we review several key topics on which our proposed approach for lifted HL-MRFs (Section 3) relies upon. We begin by reviewing probabilistic modeling and templating languages for logical MRFs, in particular HL-MRFs and PSL. Next, we review the color refinement algorithm which we use to perform lifting.

2.1 Markov Random Field, HL-MRFs & PSL

Markov random fields are an expressive formalism for defining probability distributions. A number of recent SRL approaches, notably Markov Logic (Richardson and Domingos 2006), use logic to define the potentials associated with a Markov random field. These languages translate weighted logical rules into potential functions, which are in turn used to define the Markov random field. We refer to these Markov random fields as *Logical Markov Random Fields*.

Definition 1 (Markov random field)**.** *Let $\boldsymbol{y} = y_1, y_2, ..., y_n$ be set of n random variables, $\phi = \{\phi_1, \phi_2, ..., \phi_m\}$ be m potentials describing different logical relations between variables. $\phi_i(\boldsymbol{y})$ is real valued scalar representing compliance of $\boldsymbol{y}$ with ϕ_i. Also, let $\boldsymbol{w} = w_1, w_2, \ldots, w_m$ be real valued weights associated with each potential. Then, a* Markov random field *can be defined as: $P(\boldsymbol{y}) \propto exp\left(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{y})\right)$ and a* logical Markov random field *is the same with its potentials defined through logical statements and hence, $\phi_i \in 0, 1$.*

The potentials of the MRF define how the domain behaves. These potentials can be defined using logic statements for logical MRFs. An expressive way of representing logical statements is as weighted rules, where each rule can be converted into clausal form (disjunctions of positive or negated literals). Every logical clause can be written as:

$$\left(\bigvee_{j \in I^+} y_j\right) \vee \left(\bigvee_{j \in I^-} \neg y_j\right) \tag{1}$$

where I^+ is set of positive literals that participate in the clause and I^- is a set of literals that participate in the cause with a negation (Bach et al. 2017). The most probable assignment for the variables can be found by finding the maxiumum apostieri (MAP) estimate for the distribution $\text{argmax}_{y \in \{0,1\}^n} P(\boldsymbol{y})$ as:

$$\underset{y \in \{0,1\}^n}{\text{argmax}}\, \boldsymbol{w}^T \min\left\{\sum_{i \in I^+} y_i + \sum_{j \in I^-} (1 - y_j), 1\right\} \tag{2}$$

However, this is a combinatorial optimization problem, and finding the assignment that maximizes the probability for binary random variables is equivalent to weighted MAX-SAT, a well-known NP-hard problem.

Hinge-loss MRFs A hinge-loss Markov random field (HL-MRF) is a logical MRF in which random variables are relaxed to take value in the range of $[0, 1]^n$ instead of $\{0, 1\}^n$ as in logical MRFs. In order to convert a logical MRF to a HL-MRF, we first introduce definitions for logical statements over these continuous values. Conjunction ($\tilde{\wedge}$), disjunction ($\tilde{\vee}$) and negation ($\tilde{\neg}$) are defined $y_1 \tilde{\wedge} y_2 = \max\{y_1 + y_2 - 1, 0\}$, $y_1 \tilde{\vee} y_2 = \min\{y_1 + y_2, 1\}$ and $\tilde{\neg} y = 1 - y$. The $\tilde{}$ indicates the relaxation over Boolean values. With the above relaxations the objective function in Equation 2, can be written as:

$$\underset{y \in [0,1]^n}{\text{argmin}} \sum_{i=1}^{i=m} w_i \max\{l_i, 0\} \tag{3}$$

where $l_i = 1 - \sum_{j \in I^+} y_j - \sum_{j \in I^-} (1 - y_j) = y^T x_i - c_i$, and $x_i \in \{0, 1, -1\}^n$ is a vector that determines which variables participate in the specific potential i. $x_{i,j} = 0$ implies variable y_j does not participate, $x_{i,j} = 1$ implies variable $y_j \in I^-$ and needs to be added, and $x_{i,j} = -1$ implies

variable $y_j \in I^+$ and needs to be subtracted. c_i is the constant associated with the potential. The constant is computed based on the variables that are observed and other constants in the equation.

Definition 2 (Hinge-loss energy function)**.** *Let $\boldsymbol{y} = \{y_1, y_2, ..., y_n\}$ be n random variables, $\boldsymbol{l} = \{l_1, l_2, ..., l_m\}$ be m linear constraints, and $\boldsymbol{\phi} = \{\phi_1, \phi_2, ..., \phi_m\}$ be m potentials such that, $\phi_i = (\max\{l_i, 0\})^{d_i}$, where $d_i \in \{1, 2\}$ provides a choice of two different loss functions, $d_i = 1$ (i.e., linear) and $d_i = 2$ (i.e, quadratic). For weights $\boldsymbol{w} \in \{w_1, w_2, ..., w_m\}$ a hinge-loss energy function can be defined as:*

$$f(\boldsymbol{y}) = \sum_{i=1}^{m} w_i \phi_i(y) = \sum_{i=1}^{m} w_i \max(\sum_{j=1}^{n} x_{ij} y_j - c_i, 0)^{d_i} \tag{4}$$

where $\mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1}$, and the HL-MRF is defined as: $P(\boldsymbol{y}) = \frac{1}{Z(\boldsymbol{y})} exp(-f(\boldsymbol{y}))$, where $Z(\boldsymbol{y}) = \int_{\boldsymbol{y}} exp(-f(\boldsymbol{y}))$ is a normalization factor.

A key advantage of using HL-MRFs is that by using the continuous approximation for the logic statements, inference of random variables turns into a convex optimization problem from a combinatorial problem. Additionally, this reformulation enables us to use the alternating direction method of multipliers (ADMM) (Boyd et al. 2011) to infer the variables. With ADMM, inference in HL-MRFs is scalable which allows us to perform inference on real-world datasets. The first step in solving the problem with ADMM is to form the *augmented Lagrangian function* of the problem as:

$$L(y, y_l) = \min_{y, y_l} \sum_{i=1}^{m} w_i \phi_i(y_{l,i}) + \sum_{j=1}^{n} \mathcal{X}_{[0,1]}[y_j]$$
$$\textit{s.t., } y_{l,i} = y \; \forall i \in \{1, ..., m\} \tag{5}$$

where $\mathcal{X}_{[0,1]}[y_j]$ is an indicator function which produces zero if $y_i \in [0, 1]$ and infinity otherwise, and y_l is a matrix with m rows and n columns and $y_{l,i}$ represents i^{th} row of the matrix. The augmented Lagrangian form of this is: $L(y, y_l, \alpha) = \min_{y, y_l} \sum_{i=1}^{m} w_i \phi_i(y_{l,i}) + \sum_{j=1}^{n} \mathcal{X}_{[0,1]}[y_j] + \sum_{i=1}^{m} \alpha_i^T (y_{l,i} - y) + \frac{\rho}{2} \sum_{i=1}^{m} ||y_{l,i} - y||_2^2$, where $\rho > 0$ is step size and α is a matrix of same dimension as y_l and represents the dual variables. The update equations for ADMM at iteration t are $\alpha_i^t = \alpha_i^{t-1} + \rho(y_{l,i}^{t-1} - y^{t-1})$, $y_l^t = \text{argmin}_{y_l} L(y_l, \alpha^t, y^{t-1})$, and $y^t = \text{argmin}_y L((y_l, \alpha^t, y^{t-1})$. The ADMM updates ensure that y converges to the MAP state.

Probabilistic Soft Logic Probabilistic soft logic (PSL) is a declarative language for specifying HL-MRFs. A PSL model consists of a set of weighted logical rules, e.g. Horn clauses of the form $w_r : B_1 \wedge \ldots \wedge B_m \rightarrow H$ where B_i and H are *predicates* or negated predicates.

We ground a rule r by replacing the variables with constants from data. Each $w_r \in \mathbb{R}^+ \cup \{\infty\}$ is the weight of the rule r. And each ground predicate x is coupled to an interpretation function $I(x) \in [0, 1]$ representing its truth value.

Using the relaxation of logical operators, we define the notion of *distance to satisfaction* for each ground rule in a PSL model, which for example in the Horn clause above is equal to $I(\bigwedge_{i=1}^{m} B_i) - I(H)$. A grounded PSL model induces a HL-MRF in which distance to satisfaction of each grounded rule forms a hinge-loss potential in Equation 4.

Example 1. *Consider the following PSL model with one rule that represents a transitive Knows relationship among people:*

$$w_1 : Knows(P1, P2) \wedge Knows(P2, P3) \rightarrow Knows(P1, P3)$$

Assume that in our data we have three individuals: Bob, Dan, and Elsa*, and the weight of $w_1 = 5$. Given the observations $I(Knows(Ben, Elsa)) = 1$, $I(Knows(Elsa, Dan)) = 1$, and assuming that $I(Knows(X, X)) = 0$ for every individual X, our aim is to infer truth values for the remaining atoms. The grounded model consists of four atoms that participate in four grounded rules. Let us denote the unknown truth values by variables $y_1 \ldots y_4$. The hinge-loss energy function will be: $f(\boldsymbol{y}) = 5 \max(y_1 - y_2, 0)^2 + 5 \max(-y_1 + y_2 + y_4 - 1, 0)^2 + 5 \max(y_1 - y_4, 0)^2 + 5 \max(-y_3 + 1, 0)^2$.*

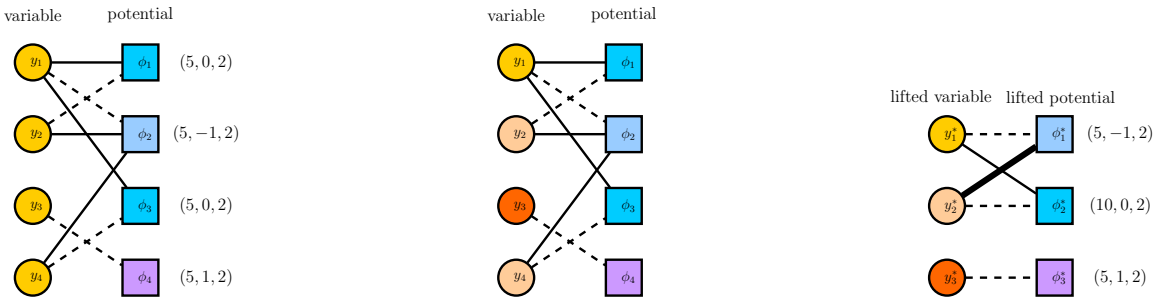
2.2 Color refinement

Color refinement is a simple algorithm to identify similar nodes in a graph (Ramana, Scheinerman, and Ullman 1994). This algorithm has efficient implementations that run in quasilinear time (Codenotti et al. 2013) and has been used in practical graph isomorphism tools and for lifted inference (Grohe et al. 2017). Color refinement is an iterative algorithm that assigns colors to nodes in a sequence of refinement rounds. For a graph $G = (V, E)$, it first initializes all nodes in V with the same color. In every refinement round, any two nodes $v, w \in V$ with the same color are reassigned to different colors if there is some color c such that v and w have a different number of neighbors with color c; otherwise no change is made. The refinement stops when the colors of all nodes before and after the refinement round remain the same. This state of the graph where the colors of nodes do not change across refinement rounds is called a *stable coloring* of the graph. Let A be the adjacency matrix of G. Then the nodes u and v have the same color in the stable coloring of G iff it holds for every color C that $\sum_{w \in C} A_{vw} = \sum_{w \in C} A_{v'w}$.

The color refinement algorithm can be generalized to weighted graphs by refining the colors based on weighted sum of the edges instead of degree. This generalization can then be extended to weighted bipartite graphs, where each part is initialized with a different color. In a stable *stable bi-coloring* of a weighted bipartite graph, the condition mentioned above holds for the weighted adjacency matrix of G.

3 Method

Graphical models generated from logical templates can manifest degrees of symmetry. In this section we introduce a method based on the color refinement algorithm to find and eliminate such symmetries in an HL-MRF energy function.



(a) Factor graph: initial coloring (b) Colored factor graph (c) Lifted factor graph

Figure 1: The factor graph of the HL-MRF model presented in Example 1. The labels of the factor nodes appear on their right side. Edge weights are represented by line style (solid: 1, dashed: -1, thick: 2).

The function obtained by this method is also a HL-MRF energy function. Preserving this form is crucial, since the efficient ADMM method introduced earlier is tailored for functions of this form. At the end of this section, we show that we can obtain the solution of the original problem by solving the lifted problem and mapping its solution back to the original space.

3.1 Lifted HL-MRFs (LHL-MRFs)

Our lifting method operates on a *factor graph*, which is a graphical representation of a HL-MRF energy function.

Definition 3 (Factor Graph)**.** *The factor graph of an HL-MRF energy function is a graph $G = (U, V, E)$ in which there is a node $u_j \in U$ for each variable y_j ($j = 1, \ldots, n$) and a node $v_i \in V$ for each potential ϕ_i ($i = 1, \ldots, m$). For each nonzero coefficient x_{ij} of variable y_j in potential ϕ_i there is an edge $e_{ij} \in E$ between u_j and v_i with the weight x_{ij}. Each node $v_i \in V$ is labeled by the tuple (w_i, c_i, d_i).*

Example 2. *The energy function of the HL-MRF in Example 1 can be represented by the factor graph in Fig. 1a.*

We will now describe a method that given the factor graph G of an energy function f, produces a potentially smaller factor graph G'. Instead of solving the MAP inference problem for f, one can solve the MAP inference problem for the function f' represented by G' and map the solution back to the variables in f.

We first assign the initial colors to the nodes of $G = (U, V, E)$. The nodes in V receive different colors based on their labels: Two nodes with labels (w_1, c_1, d_1) and (w_2, c_2, d_2) receive the same initial color iff $c_1 = c_2, w_1 = w_2$ and $d_1 = d_2$. All nodes in U receive the same color, which is different from the colors of the nodes in V. We then run the color-refinement algorithm on G, which outputs a stable bi-coloring $C_1^U, \ldots, C_p^U$ for the nodes in U and $C_1^V, \ldots, C_q^V$ for the nodes in V. To create the lifted factor graph $G' = (U', V', E')$, we first create a lifted variable node u'_k for every color class C_k^U and a lifted factor node v'_l for every color class C_l^V. Each lifted variable node u'_k and lifted factor node v'_l corresponds to a set of edges in G, namely $E_{kl} = \{e_{ij} \in E : v_i \in C_l^V, u_j \in C_k^U\}$. If E_{kl} is non-empty, we connect the nodes u'_k and v'_l in G' by an edge with the weight $(\sum_{(i,j):e_{ij} \in E_{kl}} x_{ij})/|C_l^V|$. Let $\mathcal{I} = \{i : v_i \in C_l^V\}$ and (w, c, d) be the label of some $v \in C_l^V$. We label the node $v'_l \in V'$ by the tuple $(\sum_{i \in \mathcal{I}} w_i, c, d)$.

Example 3. *The output coloring of the color refinement algorithm is shown in Fig. 1b. According to this coloring, the variables are partitioned into sets $\big\{\{y_1\}, \{y_2, y_4\}, \{y_3\}\big\}$ and the factors are partitioned into sets $\big\{\{\phi_2\}, \{\phi_1, \phi_3\}, \{\phi_4\}\big\}$. From the color classes of Example 2, we obtain the lifted factor graph of Fig. 1c which represents the function: $f(\boldsymbol{y}^\star) = 5\max(-y_1^* + 2y_2^* - 1, 0)^2 + 10\max(y_1^* - y_2^*, 0)^2 + 5\max(-y_3^* + 1, 0)^2$.*

As noted before, the expression in the above example is essentially a weighted sum of hinge functions which has the exact same form as (4). This implies that the function represented by the lifted factor graph can also be solved using ADMM. To map the solution of lifted problem back to the original space, we only need to assign the value of the representative variables of each lifting color class to all the variables in that color class.

3.2 Correctness of the method

We now show that optimizing over the lifted function produces the same objective value as optimizing the original function, and the optimal values of the variables in the original problem can be derived from their lifted counterparts. Our proof is based on an existing procedure for lifting the *Quadratic Programming* (QP) problems (Mladenov, Kleinhans, and Kersting 2017). We show that the MAP inference problem in HL-MRFs can be cast as a QP problem and that lifting this problem produces another QP which is equivalent to the function produced by our lifting method.

To write the objective function of the MAP inference in Equation 4 as a QP, we replace the max functions by constraints over auxiliary variables ψ_i:

$$\min \sum_i w_i \psi_i^{d_i} \quad \textit{s.t.,} \;\; \psi_i \geq \sum_j x_{ij} y_j - c_i \; \forall i, \quad \boldsymbol{y}, \boldsymbol{\psi} \geq \mathbf{0} \tag{6}$$

QP problems are lifted by performing the color refinement algorithm on a graph called the *coefficient graph*. We will

now explain how to construct the coefficient graph for (6) (for further details we refer to Mladenov, Kleinhans, and Kersting (2017)). The coefficient graph of (6) is the 4-tuple (U, V, Z, E) where the nodes $u_j \in U$, $v_i \in U$, and $z_i \in Z$ correspond to variable y_j, constraint i, and variable ψ_i, respectively. For each nonzero coefficient x_{ij} there is an edge with weight x_{ij} between the nodes u_j and v_i. For each constraint i, the nodes $v_i \in V$ and $z_i \in Z$ are connected by an edge with weight $-c_i$, and if $d_i = 2$ then there is a self-loop edge on z_i with the weight w_i.

Initially all nodes in $U \cup Z$ have the same color, which is not shared by any node in Z. Two nodes $v_{i_1}, v_{i_2} \in V$ receive the same initial color iff $c_{i_1} = c_{i_2}$.

After performing the color refinement algorithm on this coefficient graph, the coefficient graph of the lifted QP problem is constructed by grouping the variables and constraints of each color class together. The edge weights are aggregated in the same way as previously described in our method. The optimal value of a variable in the original QP is equal to the optimal value of its lifted counterpart.

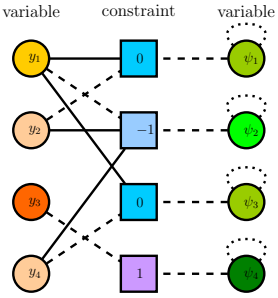


Figure 2: The colored coefficient graph of the HL-MRF model presented in Example 1. Edge weights are represented by line style (solid: 1, dashed: -1, dotted: 5).

Example 4. *The coefficient graph of the QP corresponding to Example 1 and the coloring assigned to it by the color refinement algorithm is presented in Fig. 2.*

Assume that a function f is lifted to another function f' using our proposed method. We demonstrate the correctness of our method by showing that the QP of f can be lifted to the QP of f'.

Theorem 1. *Let $G = (U^G, V^G, E^G)$ be the factor graph of an energy function of an HL-MRF, and $Q = (U^Q, V^Q, Z^Q, E^Q)$ be the coefficient graph of its QP. Then in the stable bi-colorings of G and Q, the color classes of U and V are the same.*

Proof. Assume that in the stable bi-coloring $\mathcal{C}^G$ of factor graph G, the nodes are partitioned into disjoint colors $C^G_1, \ldots, C^G_q \subseteq V^G$ and $C^G_{q+1}, \ldots, C^G_{q+p} \subseteq U^G$. Let us denote by u^G_j and v^G_i the nodes in G corresponding to variable y_j and factor ϕ_i in the HL-MRF energy function. Also let u^Q_j, v^Q_i and z^Q_i denote the nodes corresponding to variable y_j, constraint i, and auxiliary variable ψ_i in the corresponding QP problem. We will now construct a stable bi-coloring $\mathcal{C}^Q$ of the coefficient graph Q with the following properties: 1) Variable nodes have the same color classes in $\mathcal{C}^Q$ and $\mathcal{C}^G$ and constraint nodes in $\mathcal{C}^Q$ have the same color classes as factor nodes in $\mathcal{C}^G$, 2) $\mathcal{C}^Q$ is consistent with the initial coloring of Q, and 3) $\mathcal{C}^Q$ is the coarsest stable bi-coloring of the graph Q. We first construct the coloring $\mathcal{C}^Q$ and then show how the above conditions hold for it. Let $\mathcal{C}(u)$ denote the color class of u in the coloring $\mathcal{C}$. In $\mathcal{C}^Q$ we assign the colors to the nodes in U^Q, V^Q, and Z^Q based on the color classes of U^G and V^G in $\mathcal{C}^G$ as follows:

$$u^Q_{j_1} \in \mathcal{C}^Q(u^Q_{j_2}) \Leftrightarrow u^G_{j_1} \in \mathcal{C}^G(u^G_{j_2}) \tag{7}$$

$$v^Q_{i_1} \in \mathcal{C}^Q(v^Q_{i_2}) \Leftrightarrow v^G_{i_1} \in \mathcal{C}^G(v^G_{i_2}) \tag{8}$$

$$z^Q_{i_1} \in \mathcal{C}^Q(z^Q_{i_2}) \Leftrightarrow v^G_{i_1} \in \mathcal{C}^G(v^G_{i_2}) \tag{9}$$

The first property holds by definition. By definition, the nodes $u^Q_{j_1}, u^Q_{j_2} \in U^Q$ have the same initial color iff the initial colors of the nodes $u^G_{j_1}, u^G_{j_2} \in U^G$ are the same. Similarly, $v^Q_{i_1}, v^Q_{i_2} \in V^Q$ receive the same initial color iff the nodes $v^G_{i_1}, v^G_{i_2} \in V^G$ have the same initial color. Additionally, all nodes in Z^Q receive the same initial color. Hence $\mathcal{C}^Q$ is consistent with the initial coloring of Q.

To show that the coloring is stable, we need to show that the sum of edge weights connecting to the nodes in each color class is the same among all the nodes having the same color. So for each pair of variable nodes $u^Q_{j_1}, u^Q_{j_2} \in U^Q$ and color class C^Q_l it should hold that $u^Q_{j_1} \in \mathcal{C}^Q(u^Q_{j_2}) \Leftrightarrow \sum_{i:v^Q_i \in C^Q_l} x_{ij_1} = \sum_{i:v^Q_i \in C^Q_l} x_{ij_2}$. Since $\mathcal{C}^G$ is a stable coloring of G we have $u^G_{j_1} \in \mathcal{C}^G(u^G_{j_2}) \Leftrightarrow \sum_{i:v^G_i \in C^G_l} x_{ij_1} = \sum_{i:v^G_i \in C^G_l} x_{ij_2}$ which together with equation 7 proves this property. Similarly, for each pair of constraints i_1, i_2 and color class C^Q_k it should hold that $v^Q_{i_1} \in \mathcal{C}^Q(v^Q_{i_2}) \Leftrightarrow \sum_{j:u^Q_j \in C^Q_k} x_{i_1 j} = \sum_{j:u^Q_j \in C^Q_k} x_{i_2 j}$ which can be concluded from equation 8 and the fact that $v^G_{i_1} \in \mathcal{C}^G(v^G_{i_2}) \Leftrightarrow \sum_{j:u^G_j \in C^G_k} x_{i_1 j} = \sum_{j:u^G_j \in C^G_k} x_{i_2 j}$. Note that the weights of edges connecting to nodes of ψ_i are not included in these equations since ψ_i variables appear with the same coefficient in all constraints. For a pair of nodes $z^Q_{i_1}, z^Q_{i_2} \in Z^Q$ where $d_{i_1} = d_{i_2} = 1$, we should have $z^Q_{i_1} \in \mathcal{C}^Q(z^Q_{i_2}) \Leftrightarrow v^Q_{i_1} \in \mathcal{C}^Q(v^Q_{i_2})$ which trivially holds according to equation 9. Finally, when $d_{i_1} = d_{i_2} = 2$, it should hold that $z^Q_{i_1} \in \mathcal{C}^Q(z^Q_{i_2}) \Leftrightarrow v^Q_{i_1} \in \mathcal{C}^Q(v^Q_{i_2}) \wedge w_{i_1} = w_{i_2}$ which holds according to equation 9 and the fact that if $w_{i_1} \neq w_{i_2}$ then the nodes $v^Q_{i_1}, v^Q_{i_2} \in V^Q$ are initialized with different colors.

Now what remains is to show that $\mathcal{C}^Q$ is the coarsest stable coloring of graph Q, i.e., there is not another stable bi-coloring respecting the previous conditions that assigns fewer number of colors than $\mathcal{C}^Q$ to the nodes of Q. Assume that there is a stable bi-coloring $\mathcal{C}'^Q$ of Q with fewer colors than $\mathcal{C}^Q$. Then we can construct a stable bi-coloring $\mathcal{C}'^G$ for the factor graph G that respects its initial coloring, by partitioning the U^G and V^G according to the color classes of U^Q and V^Q in $\mathcal{C}'^Q$. Since partitions of U^Q and Z^Q are in one-to-one correspondence, the reduction in the number of

color classes in $\mathcal{C}'^{Q}$ can not be limited to color classes of the nodes in Z^{Q}. This means that $\mathcal{C}'^{G}$ has fewer color classes than $\mathcal{C}^{G}$, which is a contradiction. □

4 Empirical Evaluation

In this section, we evaluate our proposed lifted inference algorithm, LHL-MRF, on various real and synthetic datasets. We investigate three research questions in our experiments: **Q1:** How does lifting affect performance on real world datasets? **Q2:** How does the graph structure influence the impact of lifting? **Q3:** How much symmetry is required for lifting HL-MRFs to be effective? All experiments were run on a machine with 16GB RAM and an i5 processor. The implementations are all single-threaded. We implemented our models using the PSL open-source Java library[1]. We ground the rules using the PSL library and then run inference using our own implementation of ADMM in C++[2]. Note that PSL removes a large number of trivial symmetries during the grounding process by removing trivially satisfied rules (for further information see (Augustine and Getoor 2018)). Removing these simple symmetries ensures the extra symmetries that are obtained during our approach are non-trivial. We use Saucy[3] from the *RELOOP* library to perform color refinement (Mladenov et al. 2016).

Experiments on Real-world Data

We selected three real world datasets from different domains for which the corresponding PSL models have been used with promising results.

-**Citeseer**: This dataset includes 3312 papers in six categories, and 4591 citation links. The goal is to classify documents in a citation network. The original data comes from Citeseer . The details about the model and data can be found in Bach et al. (2017).

-**Cora**: This dataset includes includes 2708 papers in seven categories, and 5429 citation links. The goal is to classify documents in a citation network. The original data comes from Cora . The details about the model and data can be found in Bach et al. (2017).

-**Wikidata**: The dataset contains 419 families and 1,844 family trees. The goal is to perform entity-resolution on a family graph obtained form wikidata by crawling the site for familial relations. The details about the model and data can be found in Kouki et al. (2017).

To address Q1, we measure the effects of lifting on the three datasets. Figure 3 presents the number of variables and potentials of these datasets before and after lifting. We observe that there is a varying amount of symmetry in these datasets, the reduction in number of variables and potentials is about 20% in the Wikidata, 46% in the Cora, and 66% in the Citeseet dataset.

Table 1 shows the time to solve the original problem, i.e., HL-MRF, the time to solve the lifted problem, i.e., LHL-MRF (solving), the time to lift HL-MRF with the color refinement algorithm, i.e., LHL-MRF (lifting), and the end to end inference time for the lifted approach, i.e., LHL-MRF (total) or LHL-MRF in short.

Datasets	HL-MRF (in sec)	LHL-MRF (solving) (in sec)	LHL-MRF (lifting) (in sec)	LHL-MRF (total) (in sec)
Citeseer	57.4	19.8	0.39	20.19
Cora	47.7	17.5	0.53	18.03
Wikidata	636.0	463.7	112.7	576.4

Table 1: Time taken to perform inference on different datasets.

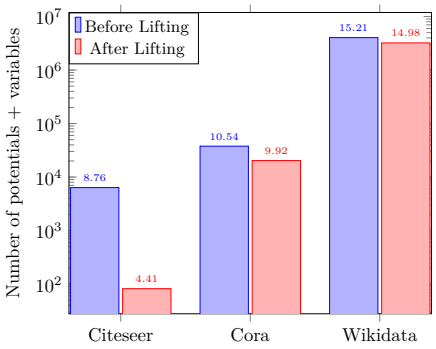


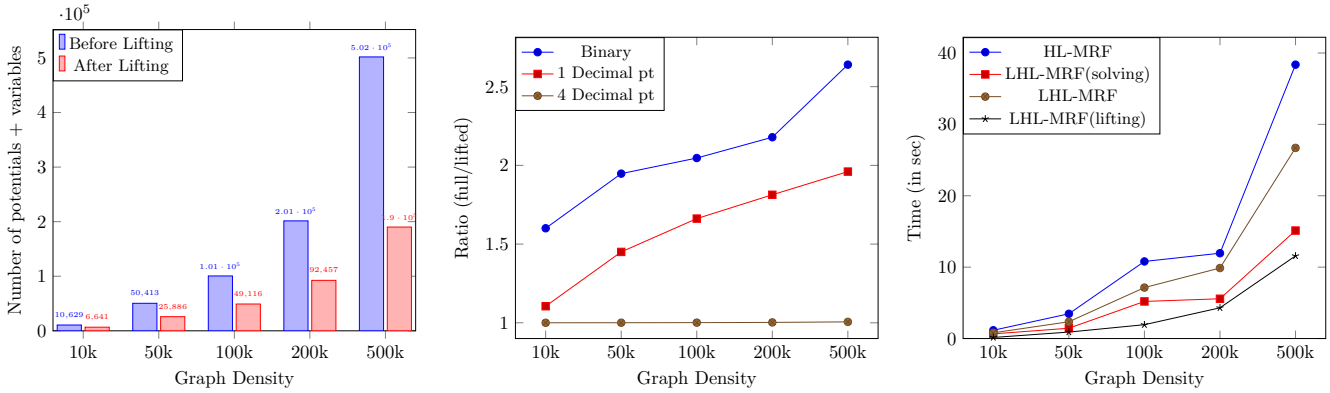
Figure 3: The number of variables and rules reduce in different amount after lifting in real-world datasets.

As expected, due to the large amount of reduction in the number of variables and potentials, there is a significant difference between the time taken for HL-MRF and LHL-MRF (solving) on all subsets. Even with a small 20% reduction in number of variables and potentials in Wikidata, we see that LHL-MRF (Solving) is 27% faster than HL-MRF and due to much higher reduction in other datasets, we see three-fold speed-ups in both the Cora and the Citeseer datasets.

However, lifting time (LHL-MRF (lifting)) must also be considered. After accounting for this, we still see that LHL-MRF is about a 10% faster in the Wikidata dataset and almost three times faster for the Cora and the Citeseer datasets when compared to HL-MRF.

Experiments on Synthetic Data To address Q2 and Q3 and better understand how symmetry is affected by graph density, we generate five different synthetic graphs. We also generate three sets of possible continuous values that the edges of the graph can take to generate different structure of neighbors in the graph. We generate the graphs for the task of node labeling with varying levels of density. We used a PSL model for the commonly used smoker example as introduced in Richardson and Domingos (2006), which describes smoking behavior among friends with the following rules:
$1.0 : Friend(A, B) \wedge Smokes(A) \rightarrow Smokes(B)$
$1.0 : Friend(A, B) \wedge \neg Smokes(A) \rightarrow \neg Smokes(B)$ This model states that if two people are friends, then either both of them smoke or neither of them do.

[1]https://github.com/linqs/psl
[2]https://github.com/linqs/srinivasan-aaai19
[3]http://vlsicad.eecs.umich.edu/BK/SAUCY



(a) For binary values, as the graph density increases, the total amount of lifting increases.

(b) For varying numbers of values, as graph density increases, the ratio of lifting varies.

(c) Comparison of inferences times for HL-MRF and LHL-MRF as graph density varies.

Figure 4: Comparison of inferences times and size of the problem for HL-MRF and LHL-MRF as graph density varies

We fix the number of users to 1000, and randomly create friendship links between these users by varying the number of edges from $10k$ to $500k$. In practice friendship is not necessarily a black-and-white matter, i.e., people can be friends to varying degrees. Hence, we consider three cases for the values of the friendship links: 1) binary values, 2) values between zero to one with one decimal point, 3) values between zero to one with four decimal points. This means that friendship links can take only two values in the first case ($\{0, 1\}$), 10 values in the second ($\{0.0, 0.1, \ldots, 1.0\}$) and 10,000 in the third case ($\{0.0000, 0.0001, \ldots, 1.000\}$). We randomly assign a label to users and keep 50% of the labels as evidence and another 50% as unknowns to be inferred. Figure 4a shows the total number of variables and potentials before and after lifting for the binary case. Figure 4b shows the ratio between the number of variables and potentials before and after lifting, for varying value ranges. We see that for the binary case, the amount of lifting is maximized and the ratio increases as the graph density increases. However, as the value range increases, the amount of lifting drops significantly, and eventually there are no symmetries to be exploited. Finally, Figure 4c presents the processing time to solve the binary case. The results indicate that using LHL-MRF gives a significant performance improvement over HL-MRF as the graph becomes denser. These results imply that there are complex trade-offs between the structure of the graph and the range of the values in the data. We utilize exact lifting and therefore, we observe that LHL-MRF performs well for finding symmetries in datasets with denser structures and smaller range of values.

Finally, to further understand how the amount of symmetry affects the overall inference time in a slightly more complex and realistic setting (yet still a synthetic dataset), we study the social affiliation dataset and the PSL model used by Bach et al. (2017) for scalability analysis. We use a dataset that contains 22k nodes and 130k edges [4].

We begin by lifting this dataset to remove all symmetry (the original dataset has less than 1% symmetry). To induce symmetry, we systematically inject the same structure to the data. This is done by duplicating every grounded rule and shuffling the data. We duplicate the grounded rules up to 10 times creating 10 subsets (named 1x, 2x, 3x..., 10x), where the 10x subset has 10 times as many potentials created by duplicating the original data i.e., the 1x subset.

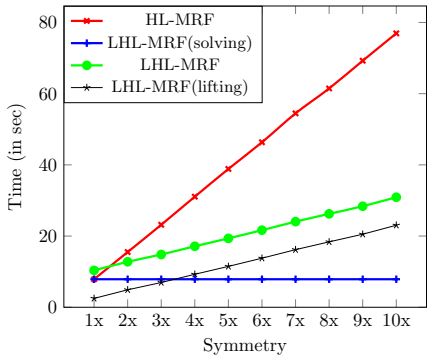


Figure 5: As symmetry increases, the gap between time solving HL-MRF and LHL-MRF increases.

Figure 5 shows the results of HL-MRF and LHL-MRF (split into LHL-MRF (solving), LHL-MRF (lifting), and LHL-MRF on all 10 subsets. For the smallest dataset (which contains no symmetry), the time required for LHL-MRF is higher than HL-MRF due to the time taken to perform lifting. However, increasing the size of the dataset from two to ten, we observe that the amount of time taken by LHL-MRF to solve the problem is getting much lower than HL-MRF. It is noticeable that as the symmetry increases, the gap between solving HL-MRF problem and LHL-MRF problems widens. Note that the inference time in LHL-MRF for all 10 subsets is the same (equal to 1x dataset), which is the flat line in Figure 5 for LHL-MRF (solving).

For the sake of completeness and to compare against other lifted inference methods, in Figure 6, we compare the performance of HL-MRF and LHL-MRF using ADMM with

[4] https://github.com/stephenbach/admm-speed-test

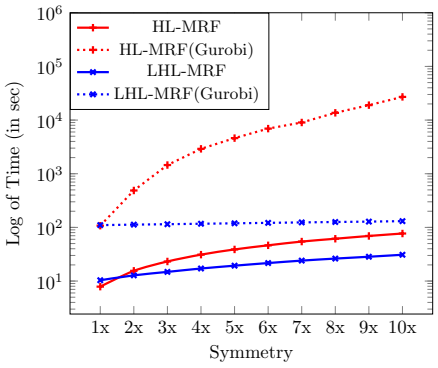


Figure 6: As symmetry increases, the gap between time solving HL-MRF and HL-MRF(Gurobi) increases exponentially. The difference between LHL-MRF and LHL-MRF(Gurobi) remains the same as the 1x dataset.

versions which use Gurobi– an off-the-shelf commercial QP solver–. We denote these methods which use Gurobi HL-MRF (Gurobi) and LHL-MRF (Gurobi). For all 10 subsets, we observe that using HL-MRF and LHL-MRF consistently and significantly outperforms HL-MRF (Gurobi) and LHL-MRF (Gurobi) respectively. We also see that this difference increases as the size of the data increases. Note that the time taken to solve using LHL-MRF (Gurobi) is similar to other lifting methods such as belief propagation. For most of the lifting methods, the time complexity grows cubicly with the number of variables in the data. However, our approach is unique and desirable as it maintains the original form of the function allowing us to use ADMM which is known to be much more scalable than other approaches (Forouzan and Ihler 2013).

To our best knowledge, the size of datasets used in other lifted inference papers are in order of 1000s of variables and potentials, whereas using ADMM in our approach allow us to easily scale to problems with millions of variables and potentials.

5 Discussion and Future Directions

We have shown that there are significant opportunities for lifted inference, even in the case where we have continuous-valued variables defined by HL-MRFs. Through empirical evaluation on real datasets, we show that the inference task for HL-MRF models can run up to three times faster by using LHL-MRFs. However, it is important to note that LHL-MRFs cannot guarantee speed-ups for all types of problems. We investigate the effects of graph density and range of real values on lifting in HL-MRFs. However, studying the characteristics of the optimization problem after lifting is left for future work. We also notice that on small sized problem, in which inference takes less than one second to finish in HL-MRFs, the overhead of lifting is noticeable, and therefore even with a huge amount of reduction in the number of variables and potentials, we cannot necessarily reduce the solving time of LHL-MRFs.

This work suggests other interesting directions for future work. First, in this work we only exploit exact symmetries, which may be hard to find in some applications. Previous work indicates approximate lifted inference can improve the performance without compromising on other metrics like precision (Sen, Deshpande, and Getoor 2009). In our setting, approximate lifting could also lead to a greater reduction in number of variables and speed up the task of inference. Second, two of the most challenging tasks in MRFs are learning the weights and the structure of the logical rules from the data. Structure learning and weight learning are often performed using a scoring function that iteratively uses a MAP state. An interesting path to explore is to employ lifted inference to make such systems more efficient.

6 Conclusion

In this paper, we introduced LHL-MRF, a novel approach to lifted inference in HL-MRFs. LHL-MRF marries the powerful ideas of lifted inference with the color refinement algorithm of Grohe et al. (2017) with the convex inference approach proposed by Bach et al. (2017), to solve large-scale graphical models described by HL-MRFs. By combining these two ideas, our method is able to reduce the number of variables and potentials in a model and perform inference efficiently on a significantly smaller optimization problem. Through empirical evaluation, we show that the inference task for HL-MRF models on relatively small real world problems can be made to run three times faster. Further, in our experiments, we investigated how varying symmetry affects the performance of LHL-MRF and we explored the impact of both structure and domain values on the efficiency of LHL-MRF.

Acknowledgement

We would like to thank Kristian Kersting for his constructive feedback and the anonymous reviewers for their helpful suggestions. This work is supported by the National Science Foundation under grant numbers CCF-1740850 and IIS-1703331, and by the US Army Corps of Engineers Research and Development Center under contract number W912HZ-17-P-0101. Behrouz Babaki is supported by a postdoctoral scholarship from IVADO through the Canada First Research Excellence Fund (CFREF) grant.

References

Aditya, S.; Yang, Y.; and Baral, C. 2018. Explicit reasoning over end-to-end neural architectures for visual question answering. In *AAAI*.

Alshukaili, D.; Fernandes, A. A. A.; and Paton, N. W. 2016. Structuring linked data search results using probabilistic soft logic. In *ISWC*.

Augustine, E., and Getoor, L. 2018. A comparison of bottom-up approaches to grounding for templated Markov random fields. In *SysML*.

Bach, S. H.; Broecheler, M.; Huang, B.; and Getoor, L. 2017. Hinge-loss Markov random fields and probabilistic soft logic. *Journal of Machine Learning Research* 18:109:1–109:67.

Beltagy, I.; Erk, K.; and Mooney, R. J. 2014. Probabilistic soft logic for semantic textual similarity. In *ACL*.

Boyd, S. P.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*.

Chavira, M., and Darwiche, A. 2008. On probabilistic inference by weighted model counting. *Artif. Intell.* 172(6-7):772–799.

Chen, P.; Chen, F.; and Qian, Z. 2014. Road traffic congestion monitoring in social media with hinge-loss Markov random fields. In *ICDM*.

Codenotti, P.; Katebi, H.; Sakallah, K. A.; and Markov, I. L. 2013. Conflict analysis and branching heuristics in the search for graph automorphisms. In *ICTAI*.

De Raedt, L., and Kersting, K. 2011. Statistical relational learning. In *Encyclopedia of Machine Learning*. Springer.

de Salvo Braz, R.; Amir, E.; and Roth, D. 2005. Lifted first-order probabilistic inference. In *IJCAI*.

Dechter, R., and Mateescu, R. 2007. AND/OR search spaces for graphical models. *Artif. Intell.* 171(2-3):73–106.

den Broeck, G. V.; Taghipour, N.; Meert, W.; Davis, J.; and Raedt, L. D. 2011. Lifted probabilistic inference by first-order knowledge compilation. In *IJCAI*.

Deng, L., and Wiebe, J. 2015. Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models. In *EMNLP*.

Ebrahimi, J.; Dou, D.; and Lowd, D. 2016. Weakly supervised tweet stance classification by relational bootstrapping. In *EMNLP*.

Forouzan, S., and Ihler, A. 2013. Linear approximation to admm for map inference. In *JMLR W&CP*, 48–61.

Getoor, L., and Taskar, B. 2007. *Introduction to statistical relational learning*. MIT.

Gogate, V., and Domingos, P. M. 2010. Exploiting logical structure in lifted probabilistic inference. In *StarAI Workshop at AAAI*.

Gridach, M.; Haddad, H.; and Mulki, H. 2017. Churn identification in microblogs using convolutional neural networks with structured logical knowledge. In *NUT workshop at EMNLP*.

Grohe, M.; Kersting, K.; Mladenov, M.; and Schweitzer, P. 2017. Color refinement and its applications. In Van den Broeck, G.; Kersting, K.; Natarajan, S.; and Poole, D., eds., *An Introduction to Lifted Probabilistic Inference*. Cambridge University Press.

Kazemi, S. M., and Poole, D. 2016. Knowledge compilation for lifted probabilistic inference: Compiling to a low-level language. In *KR*.

Kersting, K.; Ahmadi, B.; and Natarajan, S. 2009. Counting belief propagation. In *AUAI*, 277–284.

Kersting, K. 2012. Lifted probabilistic inference. In *ECAI*.

Kimmig, A.; Mihalkova, L.; and Getoor, L. 2015. Lifted graphical models: a survey. *Machine Learning* 99(1):1–45.

Kouki, P.; Pujara, J.; Marcum, C.; Koehly, L. M.; and Getoor, L. 2017. Collective entity resolution in familial networks. In *ICDM*.

Lalithsena, S.; Perera, S.; Kapanipathi, P.; and Sheth, A. P. 2017. Domain-specific hierarchical subgraph extraction: A recommendation use case. In *BigData*.

Mladenov, M.; Ahmadi, B.; and Kersting, K. 2012. Lifted linear programming. In *AISTATS*.

Mladenov, M.; Heinrich, D.; Kleinhans, L.; Gonsior, F.; and Kersting, K. 2016. Reloop: A python-embedded declarative language for relational optimization. In *DeLBP Workshop at AAAI*.

Mladenov, M.; Kersting, K.; and Globerson, A. 2014. Efficient Lifting of MAP LP Relaxations Using k-Locality. In *AISTATS*.

Mladenov, M.; Kleinhans, L.; and Kersting, K. 2017. Lifted inference for convex quadratic programs. In *AAAI*.

Poole, D. 2003. First-order probabilistic inference. In *IJCAI*.

Ramana, M. V.; Scheinerman, E. R.; and Ullman, D. 1994. Fractional isomorphism of graphs. *Discrete Mathematics*.

Richardson, M., and Domingos, P. M. 2006. Markov logic networks. *Machine Learning* 62(1-2):107–136.

Sen, P.; Deshpande, A.; and Getoor, L. 2009. Bisimulation-based approximate lifted inference. In *UAI*.

Singla, P., and Domingos, P. M. 2008. Lifted first-order belief propagation. In *AAAI*.

Sridhar, D.; Fakhraei, S.; and Getoor, L. 2016. A probabilistic approach for collective similarity-based drug-drug interaction prediction. *Bioinformatics* 32(20):3175–3182.

Wang, W., and Ku, L. 2016. Identifying chinese lexical inference using probabilistic soft logic. In *ASONAM*.