Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Lifting Relational MAP-LPs Using Cluster Signatures

Udi Apsel
Computer Science Department
Ben Gurion University of The Negev, Israel
apsel@cs.bgu.ac.il

Kristian Kersting and **Martin Mladenov**
Computer Science Department
TU Dortmund University, Germany
{kristian.kersting, martin.mladenov}@cs.tu-dortmund.de

Abstract

Inference in large scale graphical models is an important task in many domains, and in particular probabilistic relational models (e.g. Markov logic networks). Such models often exhibit considerable symmetry, and it is a challenge to devise algorithms that exploit this symmetry to speed up inference. Recently, the automorphism group has been proposed to formalize mathematically what "exploiting symmetry" means. However, obtaining symmetry derived from automorphism is GI-hard, and consequently only a small fraction of the symmetry is easily available for effective employment. In this paper, we improve upon efficiency in two ways. First, we introduce the Cluster Signature Graph (CSG), a platform on which greater portions of the symmetries can be revealed and exploited. CSGs classify clusters of variables by projecting relations between cluster members onto a graph, allowing for the efficient pruning of symmetrical clusters even before their generation. Second, we introduce a novel framework based on CSGs for the Sherali-Adams hierarchy of linear program (LP) relaxations, dedicated to exploiting this symmetry for the benefit of tight Maximum A Posteriori (MAP) approximations. Combined with the pruning power of CSG, the framework quickly generates compact formulations for otherwise intractable LPs, as demonstrated by several empirical results.

Introduction

Inference in large scale graphical models is an important task in many domains, and in particular probabilistic relational models (e.g. Markov logic networks). Such models often exhibit considerable symmetry, and it is a challenge to devise algorithms that exploit this symmetry in order to speed up inference, see e.g. (Poole 2003; Singla and Domingos 2008; Kersting, Ahmadi, and Natarajan 2009; Jha et al. 2010; Apsel and Brafman 2011; Van den Broeck et al. 2011; Noessner, Niepert, and Stuckenschmidt 2013; Ahmadi et al. 2013) and (Kersting 2012) for an overview. Recently, (Niepert 2012; Bui, Huynh, and Riedel 2013) have established connections between the automorphism group of graphical models and lifted inference. Specifically, Bui *et al.* have demonstrated how to deduce some of the symmetry available directly from the relational structure, and formulated a compact linear program (LP) for the basic relaxation of the Maximum A Posteriori (MAP) query. Their framework, however, does not provide one with efficient tools to formulate tighter MAP-LP relaxations, such as those obtained via the Sherali-Adams (SA) hierarchy. In this paper, we make two contributions that jointly allow the computation of much tighter relaxations in relational models.

The first contribution, called Cluster Signature Graph (CSG), is a graphical platform that allows an efficient detection of symmetry between clusters of variables, by projecting the relational structure of each cluster onto a graph. Consequently, isomorphic CSGs denote exchangeable clusters, that is – clusters that are mapped one onto the other in a permutation belonging to the automorphism group.

The second contribution, inspired by the work of (Mladenov, Globerson, and Kersting 2014), is a framework which formulates compact MAP-LPs of the SA hierarchy. At its core, the framework relies on an oracle to provide all non-exchangeable clusters within a given size range. The above requirement is efficiently met by generating non-isomorphic CSGs using automated graph generation tools. Since the generated graphs are small, and the procedure for producing such graphs involves no analysis of the ground graphical model, the combination of our two contributions produces compact MAP-LPs of the SA hierarchy very quickly.

Further, we discuss the integration of evidence on unary and binary atoms into the framework, and finally, validate our new approach with several empirical results, putting special emphasis on the challenging class of transitive relational models. From the empirical results, we highlight several lifting of the SA hierarchy up to level 6, in non-trivial models.

We begin with an elaborated background section, followed by the presentation of our MAP approximation framework. Only then do we proceed to the CSG introduction.

Background

MRF and MAP Estimation

A Markov Random Field (*MRF*) $\mathcal{M}$ is a probabilistic graphical model, consisting of a set of random variables $x = \{x_1, \ldots, x_n\}$, and a set of *factors*. A factor is a pair (θ_f, x_f), which represents a function $\theta_f : \mathrm{range}(x_f) \to \mathbb{R}$, mapping from the joint assignment range of variables $x_f \subseteq x$, to the reals. The joint distribution function of MRFs is given by

$$Pr(x = \hat{x}) = \frac{1}{Z} \prod_f \exp\left(\theta_f(\hat{x}_f)\right) \quad (1)$$

where $\hat{x}$ is a joint assignment to all x variables, and $\hat{x}_f$ is the respective joint assignment to all x_f variables under $\hat{x}$. Z here denotes a normalization constant called the *partition function*. One common query in probabilistic models is the Maximum A Posteriori (*MAP*) estimation query, which seeks to obtain a joint assignment that maximizes the joint distribution. The unnormalized maximization task can be expressed in the form of sum of logarithms, as follows.

$$\text{MAP}(\mathcal{M}) = \max_x \sum\nolimits_f \theta_f(x_f) \tag{2}$$

In this paper, we will be focusing on obtaining an upper bound on the maximal unnormalized log distribution in relational probabilistic models. MAP assignments corresponding to this task may be obtained via rounding schemes, e.g. (Ravikumar, Agarwal, and Wainwright 2010). Optimizing the latter for the benefit of relational environments is an interesting challenge, which we leave to further study.

MAP Relaxation via Linear Programming

Methods for approximating the NP-hard MAP query (Shimony 1994) have become a subject for extensive study in various fields of computer science. Of the many approaches (Yedidia, Freeman, and Weiss 2003; Sontag, Globerson, and Jaakkola 2011), we build our work on the framework of Linear Programming (LP), where the MAP computation problem is translated into an Integer Linear Program (ILP) (Schrijver 1998), and relaxed to an LP by lifting the integrality constraints imposed on the variables. Formally, an LP is a maximization problem over a set of real-valued variables $\mu = \{\mu_i\}_i$, as follows.

$$\max_\mu \sum\nolimits_i c_i \mu_i \qquad \text{s.t.} \qquad \sum\nolimits_i a_{ij} \mu_i \leq b_j \qquad \forall j \tag{3}$$

where c_i, b_j and a_{ij} denote real-valued coefficients. The polynomial-time complexity of LPs (Khachiyan 1979; Karmarkar 1984) lends itself to fast approximation methods, and can be found suitable for large domains, as in our setting.

For ease of presentation, our formulation, from this point on, will be adapted to triplewise MRFs and Boolean variables. This chosen formulation should suit our running examples throughout the paper. Nevertheless, the method we introduce can be applied to any MRF, of any variable assignment range. Given a triplewise MRF $\mathcal{M}$ over the set of random variables $x = \{x_i, \ldots, x_n\}$ and factors $F = \{(\theta_f, x_f)\}_f$, we define a MAP linear program as follows.

For each subset of indices $\mathcal{I}$ taken from $\{1, \ldots, n\}$ of size $1 \leq |\mathcal{I}| \leq 3$, let $\mu_\mathcal{I}$ denote a vector of variables of size $2^{|\mathcal{I}|}$. A notation $\mu_{ijk}(x_i, x_j, x_k)$ will be used to describe a specific variable in vector $\mu_\mathcal{I}$ corresponding to the subset $\mathcal{I} = (i, j, k)$ and entry $(x_i, x_j, x_k) \in \{0, 1\}^3$. Additionally, let $\mathcal{I}_F$ denote the set of all ordered indices for which there exists a factor f with a matching variables scope x_f, and let θ_{ijk} denote the log probability table of a factor whose variables scope is (x_i, x_j, x_k). The MAP linear program (*MAP-LP*) is defined as the optimization problem in Figure 1. Note that permutations of $\mathcal{I}$ induce μ vectors that represent the same LP variables. For instance, the LP variable $\mu_{ij}(x_i, x_j)$ is equal to $\mu_{ji}(x_j, x_i)$, for any i, j and assignments to x_i, x_j.

$$\max_\mu \sum_{(i)\in\mathcal{I}_F} \langle \theta_i, \mu_i \rangle + \sum_{(i,j)\in\mathcal{I}_F} \langle \theta_{ij}, \mu_{ij} \rangle + \sum_{(i,j,k)\in\mathcal{I}_F} \langle \theta_{ijk}, \mu_{ijk} \rangle$$

$$\begin{aligned}
\text{s.t.} \quad & \textstyle\sum_{x_i} \mu_i(x_i) = 1 && \forall (i) \in \mathcal{I}_F, \\
& \textstyle\sum_{x_j} \mu_{ij}(x_i, x_j) = \mu_i(x_i) && \forall (i,j) \in \mathcal{I}_F, \\
& \textstyle\sum_{x_i} \mu_{ij}(x_i, x_j) = \mu_j(x_j) && \\
& \mu_{ijk}(x_i, x_j, x_k) \geq 0 && \forall (i,j,k) \in \mathcal{I}_F, \\
& \textstyle\sum_{x_k} \mu_{ijk}(x_i, x_j, x_k) = \mu_{ij}(x_i, x_j) && \\
& \textstyle\sum_{x_j} \mu_{ijk}(x_i, x_j, x_k) = \mu_{ik}(x_i, x_k) && \\
& \textstyle\sum_{x_i} \mu_{ijk}(x_i, x_j, x_k) = \mu_{jk}(x_j, x_k) &&
\end{aligned}$$

where ,

$$\langle \theta_{ijk}, \mu_{ijk} \rangle = \sum_{(x_i, x_j, x_k)} \theta_{ijk}(x_i, x_j, x_k) \cdot \mu_{ijk}(x_i, x_j, x_k)$$

Figure 1: MAP-LP for triplewise MRFs

For this reason, we sometimes ignore the ordered nature of $\mathcal{I}$ and refer to its respective set of $x_\mathcal{I}$ variables as a cluster.

Modern LP solvers are capable of solving MAP-LP instances for some very large models. However, depending on a given instance of a problem, the quality of approximation may prove to be insufficient. Thus, it is common to employ strategies that produce tigher results. One often used strategy is to identify violations of cycle inequalities (Sontag and Jaakkola 2007), and add constraints that remove these solutions from the LP's feasible region. An adaptation of this method to relational probabilistic models was recently introduced in (Bui, Huynh, and Riedel 2013). A different strategy, albeit considered intractable in large or even moderately sized models, is elevating the MAP approximation problem onto high levels of *lift and project hierarchies*. As shown in (Sontag 2010), such approach, if tractable, produces tighter bounds compared with the aforementioned strategy.

The Sherali-Adams Hierarchy of MAP Relaxation

Lift and project hierarchies are frameworks for incrementally tightening the LP's feasible region of a relaxed ILP, starting from loose approximations and ending in exact solutions. At each level of the hierarchy, new variables and constraints are introduced, thereby lifting the dimension of the computational problem. A solution is obtained by projecting the result back onto the smaller dimension of the original problem. Typically, lift and project methods (Sherali and Adams 1990; Lovász and Schrijver 1991; Lasserre 2002) add variables which correspond to clusters of variables taken from the original problem. In this work, we focus on the the Sherali-Adams (SA) hierarchy, which is a natural extension to MAP-LP. The principles we demonstrate, however, apply to other hierarchies as well.

In order to obtain a k-level SA linear program, called MAP-LP$_k$, we define the set of constraints $SA_\mathcal{I}$.

$$SA_\mathcal{I} \equiv \begin{cases} \mu_\mathcal{I}(x_\mathcal{I}) \geq 0 & \forall i \in \mathcal{I},\ \forall x_\mathcal{I} \in \{0,1\}^{|\mathcal{I}|} \\ \sum_{x_i} \mu_\mathcal{I}(x_\mathcal{I}) = \mu_{\mathcal{I}\setminus\{i\}}(x_{\mathcal{I}\setminus\{i\}}) & \end{cases}$$

For later references, let the constraint $\mu_\mathcal{I}(x_\mathcal{I}) \geq 0$ be called a **non-negativity constraint**, and let the constraint $\sum_{x_i} \mu_\mathcal{I}(x_\mathcal{I}) = \mu_{\mathcal{I}\setminus\{i\}}(x_{\mathcal{I}\setminus\{i\}})$ be called a **local constraint**.

MAP-LP$_k$ can be obtained by adding $\mu_{\mathcal{I}}$ vectors and respective $SA_{\mathcal{I}}$ constraints to MAP-LP$_{k-1}$, for each ordered subset of indices $\mathcal{I}$ of size k taken from $\{1, \ldots, n\}$. As a starting point, MAP-LP$_3$ is defined according to Figure 1, complemented by local constraints on all triples and pairs of variables. MAP-LP$_4$ can then be obtained from MAP-LP$_3$, by introducing $\mu_{\mathcal{I}}$ variables corresponding to all clusters of size 4, adding all constraints of the form $\sum_{x_l} \mu_{ijkl}(x_i, x_j, x_k, x_l) = \mu_{ijk}(x_i, x_j, x_k)$.

Exploiting Symmetry in Linear Programs

The SA hierarchy can quickly generate large and intractable LPs, even for levels as low as 3. Fortunately, LPs that are highly symmetrical can have their dimension substantially reduced. This aspect was originally discussed for ILPs and Semi-Definite Programs, see e.g. (Margot 2010; Gatermann and Parrilo 2004), and has recently been adapted to LPs using a quasi-linear time preprocessing step and introduced to the domain of relational probabilistic models (Mladenov, Ahmadi, and Kersting 2012). We will now formalize this notion of symmetry for the benefit of this work.

Let $\mathcal{L}$ denote an LP, and let $\{\mu_i\}_i$ denote $\mathcal{L}$'s variables.

Definition 1. *μ_i and μ_j are called **exchangeable** if there exists a permutation on $\mathcal{L}$'s variables that maps μ_i onto μ_j and yields the exact same LP. All variables exchangeable with one another form a set called an **orbit**. The set of all orbits in $\mathcal{L}$ is called $\mathcal{L}$'s **orbit partition**.*

Theorem 1. *(Bödi, Herr, and Joswig 2013) Let $\bar{\mu}_\rho$ denote the set of variables in orbit ρ. Then, there exists a solution for $\mathcal{L}$ under which $\forall \mu_i, \mu_j \in \bar{\mu}_\rho : \mu_i = \mu_j$, for each orbit ρ in $\mathcal{L}$'s orbit partition.*

From theorem 1, it immediately follows that the dimension of the LP can be reduced by defining a variable μ_ρ for each orbit, and replacing each variable μ_i whose orbit is ρ with μ_ρ. However, as the LP becomes substantially bigger and more complex in structure with each level of the SA hierarchy, such direct approach is unlikely to be feasible. In (Ostrowski 2012), the issue is tackled by incrementally generating, row by row, a compact MAP-LP$_k$, instead of generating the original MAP-LP$_k$ and only then reducing its dimension. The method is able to generate MAP-LP$_3$ on many problem instances, yet the time required to generate the LP approaches hundreds of seconds in small problem instance, and climbs up to ten of thousands seconds in problem instances of no more than a few hundred of variables and constraints. LPs for relational models, however, can easily reach thousands if not millions of variables and constraints, and may be found unsuitable for this type of approach.

Relational MRF

Probabilistic Relational Models (*PRM*) are representations of probabilistic models using the first-order predicate logic. Two commonly used PRMs are the *parfactor model* (Poole 2003) and *Markov logic network* (Richardson and Domingos 2006), and each can be considered a relational representation of an MRF. Here, we adapt a slightly modified representation of the parfactor model, as to allow a consistent set of notations for both relational and non-relational cases. We include a brief representation of the model, and refer to (de Salvo Braz, Amir, and Roth 2005) for a formal introduction.

A *domain* is a set of constants, called *domain objects*, that represent distinctive entities in the modeled world, e.g. $\{Alice, Bob, Carol, \ldots\}$. A logical variable (*lvar*) is a variable whose assignment range is associated with some domain. An *atom* is an atomic formula of the form $p(t_1, \ldots, t_n)$, where the symbol p is called a *predicate*[1], and each term t_i is either a domain object or an lvar. A *ground atom* is an atom whose t_i terms are all domain objects. Non-ground atoms are collections of ground atoms, all sharing the same assignment range, and describing a certain property of an individual (e.g. smoker) or some relation between individuals (e.g. friendship). A *ground substitution* $\{X_i/o_i\}_i$, is the replacement of each lvar X_i with a domain object o_i.

The parfactor model $\mathcal{M}^r$ (aka *relational MRF*) is a collection of relational factors, called *parfactors*. A parfactor is a tuple (θ, A, R), consisting of a function $\theta : \mathrm{range}(A) \to \mathbb{R}$, an ordered set of atoms A, and a set of constraints R imposed on A's lvars. Grounding a parfactor is done by applying all ground substitutions that are consistent with R, resulting in a collection of factors. The ground atoms then serve as random variables in the ground MRF. A notation $\theta(A \mid R)$ is commonly used to denote a parfactor. For example, parfactor $\theta(smokes(X), smokes(Y), friends(X, Y) | X \neq Y)$, whose ground instances in the domain $\{Alice, Bob\}$ are

- $\theta\big(smokes(Alice), smokes(Bob), friends(Alice, Bob)\big)$
- $\theta\big(smokes(Bob), smokes(Alice), friends(Bob, Alice)\big)$

We restrict our attention to shattered (de Salvo Braz, Amir, and Roth 2005) models, consisting of inequality constraints of the form $X \neq Y$ only. Additionally, such inequality constraints will be imposed on each pair of lvars X, Y where, in their absence, a ground factor with multiple entries of the same ground atom may be produced. For instance, $\theta(p(X), p(Y))$ may produce a ground factor $\theta(p(o_1), p(o_1))$, and will therefore be split into two parfactors: $\theta(p(X), p(Y) | X \neq Y)$ and some $\theta'(p(X))$.

Lifting the Sherali-Adams Hierarchy

Before introducing our framework, which substantially reduces the size of relational MAP linear programs induced by the SA hierarchy, we make the following observation. If two clusters of MRF variables are symmetrical, then the vectors of MAP-LP variables corresponding to these clusters are symmetrical as well. More formally, let $x_{\mathcal{I}}$ and $x_{\mathcal{J}}$ denote two clusters of variables taken from MRF $\mathcal{M}$.

Definition 2. *$x_{\mathcal{I}}$ and $x_{\mathcal{J}}$ are called **exchangeable** if there exists a permutation π on $\mathcal{M}$'s random variables that yields the exact same MRF, and under which each variable $x_i \in x_{\mathcal{I}}$ is mapped onto a distinctive $x_j \in x_{\mathcal{J}}$.*

Observation 1. *If $x_{\mathcal{I}}$ and $x_{\mathcal{J}}$ are exchangeable, then the LP variables in vectors $\mu_{\mathcal{I}}$ and $\mu_{\mathcal{J}}$ of a respective MAP-LP$_k$ are exchangeable.*

Thus, if we obtain indices of only non-exchangeable MRF clusters, then we can form a compact MAP-LP representing

[1] Although the term predicate is used, atoms are not restricted to Boolean assignments.

Algorithm 1: GENERATECOMPACTMAPLP$_k$

Input: Relational MRF $\mathcal{M}^r$, an integer k
Output: MAP-LP$_k$ of reduced dimension

objective $= 0$, constraints $= \{\}$
for $d = 1$ ***to*** k **do**
  for $\mathcal{I} \in$ CANONICALCLUSTERS($\mathcal{M}^r$,d) **do**
    Let $\mu_{\mathcal{I}}$ be a vector of LP variables of size 2^d
    Add **non-negativity constraints** on $\mu_{\mathcal{I}}$
    $(\theta_{\mathcal{I}}, x_{\mathcal{I}}) =$ CANONICALFACTOR($\mathcal{M}^r, \mathcal{I}$)
    Add $\#\mathcal{I} \cdot \langle \theta_{\mathcal{I}}, \mu_{\mathcal{I}} \rangle$ to the objective
    for $i \in \mathcal{I}$ **do**
      $\mathcal{I}' =$ CANONICALCLUSTER($\mathcal{M}^r, \mathcal{I} \setminus \{i\}$)
      Add **local constraints** between $\mu_{\mathcal{I}}$ and $\mu_{\mathcal{I}'}$
return *objective, constraints*

the k-level of the Sherali-Adams hierarchy. The unique representative clusters will be called *canonical clusters* henceforth, and their generation will be, at this point, associated with an oracle which supports the following queries:

1. CANONICALCLUSTERS($\mathcal{M}^r, d$) – generating the indices of all canonical clusters of size d in the relational MRF.
2. CANONICALCLUSTER($\mathcal{M}^r, \mathcal{I}$) – retrieving the indices of a canonical cluster which is exchangeable with $x_{\mathcal{I}}$.
3. CANONICALFACTOR($\mathcal{M}^r, \mathcal{I}$) – retrieving a ground factor $(\theta_{\mathcal{I}}, x_{\mathcal{I}})$ associated[2] with the canonical cluster $x_{\mathcal{I}}$.
4. $\#\mathcal{I}$ – retrieving the number of clusters exchangeable with $x_{\mathcal{I}}$[3] in the relational MRF.

Algorithm 1 introduces the framework for generating a compact MAP-LP, making use of the oracle's capabilities. The algorithm generates all LP variables corresponding to canonical clusters of various sizes, injects non-negativity and locality constraints, and updates the linear objective function with the matching log-potential tables. Note that the effect of the log-potential is multiplied by $\#\mathcal{I}$ to compensate for the effective removal of exchangeable variables from the LP. Also note that local constraints cannot be properly formed without invoking the oracle's CANONICALCLUSTER query, since having $\mathcal{I}$ as the set of indices of a canonical cluster, does not guarantee the same for $\mathcal{I} \setminus \{i\}$. Finally, we define $\mu_\emptyset$ as a constant whose value is 1, thereby allowing local contraints to be added for clusters of size 1.

Exchangeable Clusters in the Relational MRF

Algorithm 1 establishes a clear motivation for shifting the attention from symmetry analysis in the LP domain to the domain of the original graphical model. In particular, we're interested in capturing symmetry of clusters, as induced by the first-order formulation of the relational MRF. We begin with a known result (Bui, Huynh, and Riedel 2013).

Theorem 2. *Let π_o be a renaming permutation on the domain objects of $\mathcal{M}^r$. Let $x_{\mathcal{I}}$ and $x_{\mathcal{J}}$ be clusters of ground atoms, mapped one onto the other under π_o. Then, $x_{\mathcal{I}}$ and $x_{\mathcal{J}}$ are exchangeable.*

We demonstrate the above with an example. Let $\pi_o = \{o_1 \rightarrow o_2, o_2 \rightarrow o_1\}$ be a renaming permutation on domain objects $\{o_1, o_2\}$. Then, the two clusters of ground atoms[4] $\{r_{(1,2)}, p_2\}$ and $\{r_{(2,1)}, p_1\}$ are necessarily exchangeable. Nevertheless, renaming permutations cannot rule out exchangeability in all cases. For instance, clusters $\{p_1, q_2\}$ and $\{p_1, q_1\}$ are indeed exchangeable in parfactor $\theta(p(X), q(Y))$, although they do not conform to any renaming permutation. In the following subsection we present a generalization to the renaming permutation criterion, allowing us to capture these cases of exchangeability as well.

Cluster Signature Graph (CSG)

Our approach is based on a reduction from a cluster of ground atoms to a graph, such that if two graphs are isomorphic, then their respective clusters are exchangeable. For that purpose, we define the *Cluster Signature Graph* (CSG). Intuitively, the CSG depicts which ground atoms in the respective cluster must "move together" in any permutation which aims to preserve the structure of the relational MRF. For instance, a structure preserving permutation in parfactor $\theta(r(X,Y), p(X))$ would be required to coordinate the images of ground atoms $\{r_{(1,2)}, p_1\}$ in a way that guarantees identical substitutions on lvar X. Consequently, their images could be $\{r_{(3,5)}, p_3\}$, but never $\{r_{(3,5)}, p_4\}$. Before formalizing this restriction, we make a distinction between substitutions in different positions of the ground atom. In $r_{(3,5)}$, a substitution X/o_3 was applied on the lvar in position 1, whereas a substitution Y/o_5 was applied in position 2. With this, we are ready for the following definition.

Definition 3. *Ground atoms x_i and x_j are **substitution coupled** in positions ω_i and ω_j, respectively, if (1) they share a ground factor originating from a substitution on the same lvar in the respective ω positions, or (2) there exists a ground atom x_k that is substitution coupled with both in their respective ω positions.*

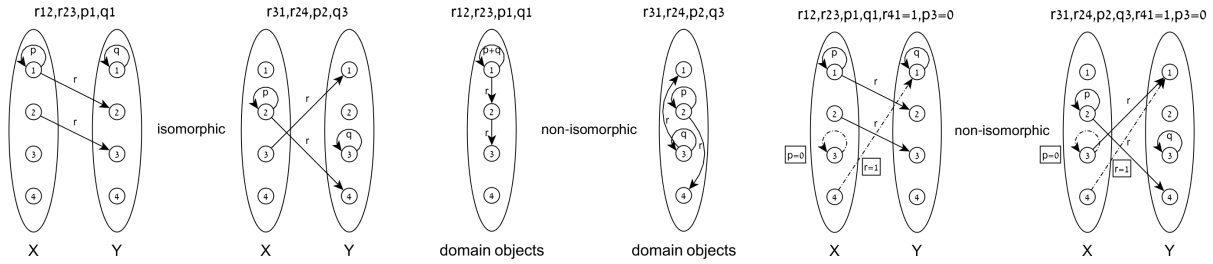
Example: In parfactor $\theta(r(X,Y), p(X))$, the ground atoms $r_{(1,2)}$ and $r_{(1,3)}$ are substitution coupled in positions 1 and 1, respectively, since they each share a ground factor with p_1. However, $r_{(2,1)}$ and $r_{(3,1)}$ are not substitution coupled, although they share the same substitution on lvar Y.

In order to provide a full picture on substitution coupling, the CSG must also integrate inequality constraints found in the relational MRF. One way to achieve the above, is to determine substitution coupling in a slightly modified, constraint-less model, and project the coupling property onto the original model, as follows. For every constraint $X \neq Y$ found in a parfactor consisting of atom $r(X,Y)$, $p(X)$ or $q(Y)$, add a respective atom $r(Y,X)$, $p(Y)$ or $q(X)$ to the parfactor and remove the constraint. The substitution

[2] An all zero factor is returned in case of no such association.

[3] In the relational setting, clusters corresponding to ground instances of the same relational factor are necessarily exchangeable, if the parfactor consists of inequality constraints on all pairs of lvars. If so, $\#\mathcal{I}$ is simply the number of ground instances of the parfactor. When not assuming this form, it is possible for two (or more) canonical clusters to be associated with the same parfactor. $\#\mathcal{I}$ will then be obtained combinatorically.

[4] Let p_i and $r_{(i,j)}$ denote $p(o_i)$ and $r(o_i, o_j)$, respectively.



(a) CSGs (b) RCSGs (c) CSGs with Evidence (broken lines)

Figure 2: Cluster Signature Graphs in $\theta(r(X,Y), p(X), q(Y))$

coupling of the modified model now requires a "coordination" between lvars X and Y, as they occupy the same position in one of the atoms, and thus fulfills the original requirement. We now proceed with the main definition.

Definition 4. *The CSG of cluster $x_{\mathcal{I}}$ in relational MRF $\mathcal{M}^r$, is a directed colored multigraph $G = (V, E, C)$, where V is a set of vertices, E is a set of directed edges and C is a coloring function, mapping each edge to a color. Edges and colors in the CSG are defined as follows.*

1. *For each $x_i \in x_{\mathcal{I}}$ originating from a unary predicate p, let G contain a node u and a loop carrying the color 'p'.*
2. *For each $x_i \in x_{\mathcal{I}}$ originating from a binary predicate r, let G contain nodes u and v to represent x_i's positions, and a directed edge $u \to v$ carrying the color 'r'.*
3. *Let all substitution coupled nodes in G be merged into a single node, without removing any of the incident edges.*

Theorem 3. *If the CSGs of $x_{\mathcal{I}}$ and $x_{\mathcal{J}}$ in $\mathcal{M}^r$ are isomorphic, then $x_{\mathcal{I}}$ and $x_{\mathcal{J}}$ are exchangeable.*

Proof outline.

Let $g_{\mathcal{I}}$ and $g_{\mathcal{J}}$ denote two isomorphic CSGs of $x_{\mathcal{I}}$ and $x_{\mathcal{J}}$, respectively. It can be shown that for every ground atom x_i added to CSG $g_{\mathcal{I}}$, there exists a respective ground atom x_j that, once added to CSG $g_{\mathcal{J}}$, induces the same isomorphism as before, but for the additional mapping between the new graph properties. Once both graphs consist of all ground atoms, the mapping between the edges of the graphs serves as a structure preserving permutation in the relational MRF. Hence, $x_{\mathcal{I}}$ and $x_{\mathcal{J}}$ are exhangeable. □

Example: Let $\mathcal{M}^r$ consist of $\theta(r(X,Y), p(X), q(Y))$ and a domain size 4, let $x_{\mathcal{I}} = \{r_{(1,2)}, r_{(2,3)}, p_1, q_1\}$ and let $x_{\mathcal{J}} = \{r_{(3,1)}, r_{(2,4)}, p_2, q_3\}$. The two clusters are exchangeable, since there exists a permutation which maps one to the other while preserving the MRF structure. That permutation is $\pi = \{r_{(1,2)} \leftrightarrow r_{(2,4)}, p_1 \leftrightarrow p_2, q_2 \leftrightarrow q_4, r_{(2,3)} \leftrightarrow r_{(3,1)}, p_2 \leftrightarrow p_3, q_3 \leftrightarrow q_1\}$. See Figure 2a for illustration.

Renaming Signatures As mentioned earlier, renaming permutations constitute a stricter form of exchangeability compared with CSGs. A graphical depiction of this form, called *Renaming Cluster Signature Graph* (RCSG), can be obtained from existing CSGs by projecting their content onto the axis of domain objects (see Figure 2b).

Generating Canonical Clusters

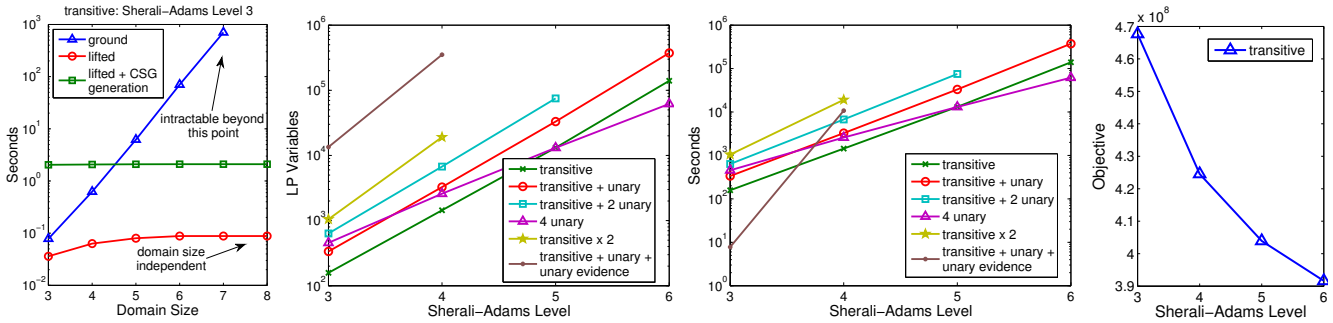
Since isomorphic CSGs denote exchangeable clusters, canonical clusters can be obtained, in principle, by generating valid CSGs and filtering out isomorphic instances. This approach, unfortunately, cannot be easily implemented due to the complex rules that constitute a valid CSG. A more practical approach is to generate non-isomorphic instances of the simpler RCSG, and filter-out further, if one wishes, by converting flat RCSG representations into CSGs.

Generating non-isomorphic instances of graphs up to k edges is clearly harder than determining isomorphism in graphs of order k. However, one must keep in mind that this procedure involves generation of small local graphs, and does not involve any processing of the actual ground (and typically large) graphical model. In fact, this procedure is entirely domain size independent. Fortunately, there are commonly available tools dedicated to this task, producing graph instances very quickly. For instance, we were able to obtain the entire data required to lift MAP-LP$_6$ in a transitive model, in less than 23 seconds, which is only a fraction of the time the LP engine takes to solve that compact LP (more than 1000 seconds). Much like knowlege compilation, generation of graphs can be carried out once, offline, and serve all models consisting of similar predicate content.

Canonical Clusters in Presence of Evidence

As it turns out, the platform of CSG is almost seamlessly extended for handling evidence. Given cluster $x_{\mathcal{I}}$ and evidence e (a set of ground atoms paired with value assignments), a new CSG is defined on the union of $x_{\mathcal{I}}$ and e, where the substitution coupling property of ground atoms in e is defined as if they contained no evidence. Edges denoting evidence are colored according to the predicate symbol and the associated value assignment, e.g. $r = 1$, $p = 0$ etc. Exchangeability, as before, is determined via an isomorphism test on the CSGs.

The set of canonical clusters is now obtained by generating all possible compositions between (a) the evidence and (b) all canonical clusters obtained by ignoring the evidence, followed by the filtration of isomorphic instances. In Figure 2c, we see two different compositions of an "evidence free" canonical cluster with constant evidence, resulting in two clusters that are no longer symmetrical. The number of canonical clusters therefore becomes highly dependent on



(a) Ground vs. Lifted (b) Number of Lifted MAP-LP Variables (c) Running Times for Lifted MAP-LP (d) Tightened Objective

Figure 3: Empirical Results for Lifted MAP-LP

the type of the evidence. Evidence on binary predicates, even in small portions, can very quickly break the symmetry of the model up to intractability.

In contrast, evidence on unary predicates results in LPs which are size indepednent in both the global domain-size and the amount of evidence. The reason for this very convenient integration, is that the composition between "evidence-free" canonical clusters and the constant evidence partitions the CSG into two disconnected components: (a) one that only carries unary evidence, and (b) one that paints the nodes of the canonical cluster with the unary evidence occupying the shared nodes. Since the evidence is constant, the content of evidence in (b) determines (a) as well. Further, nodes in (a) are graphically indistinguishable, but for the colors of the atoms occupying the nodes. These inherent properties reduce the CSG isomorphism test to a comparison between the (b) components of the CSGs, ignoring (a) altogether.

With the introduction of evidence, a single parfactor in the compact LP may be matched with several canonical clusters, all assuming similar shape but differ in evidence nodes. For instance, let model $\theta(p(X), r(X, Y))$ consist of evidence $p_1 = 0$, and let $\{p_2, r_{(2,5)}\}$ and $\{r_{(1,3)}\}$ denote two different canonical clusters. The first cluster is matched with potential θ. The second is matched with the potential θ conditioned on $p_1 = 0$. The number of clusters exchangeable with each canonical cluster ($\#\mathcal{I}$) can be determined combinatorically.

Implementation and Empirical Results

We demonstrate the framework with the challenging task of lifting MAP-LP$_k$ in transitive models. Let parfactor $\theta(r(X, Y), r(X, Z), r(Y, Z)|X \neq Y, X \neq Z, Y \neq Z)$ be called, henceforth, the *transitive model*. We follow Algorithm 1, as follows. (1) Canonical clusters are obtained by generating all non-isomorphic instances of directed graphs with up to k edges, using the *nauty* (McKay and Piperno 2014) software package, where all graphs assume a *canonically labeled* form. (2) A canonical cluster of size 3 consisting of nodes u, v, w and edges $(u, v), (u, w), (v, w)$, is matched with the parfactor, and a linear expression involving its respective μ variables is added to the objective. (3) For each canonical cluster of size d, we obtain subset clusters of size $d - 1$ for which local constraints are added, by removing edges (one each time) from the d size cluster's graph and obtaining a matching canonical labeling (nauty).

We conducted experiments on several models, in which table entries were set at random, with explicit discouragement of uniform assignment entries (Apsel and Brafman 2012), as to avoid trivial MAP solutions. The LP engine we used is the GNU Linear Programming Kit (GLPK) simplex solver. Figure 3a depicts a comparison between ground and lifted MAP-LP$_3$ for the transitive model, where we see that the ground LP could not solve instances beyond domain size 7. The lifted MAP-LP is slightly more compact in smaller domains, since canonical clusters are restricted to include no more nodes than the domain size. Beyond that, lifted MAP-LP is domain size independent. Figures 3b and 3c depict the number of LP variables and the running time of MAP-LPs for the following models: (1) Transitive; (2) Transitive with an additional parfactor and one unary atom; (3) Transitive with three additional parfactors and two unary atoms; (4) Four unary atoms and three parfactors; (5) Two binary atoms with two transitive parfactors; (6) Model no. 2 with unary evidence on 50% of the atoms. Figure 3d depicts the tightening of the objective with each level of the SA hierarchy. To the best of our knowledge, we present the highest lifting of the SA hierarchy in relational models, to date.

Conclusions

We presented a new framework for relational MAP-LP, based on the Sherali-Adams hierarchy, which exploits symmetry between clusters to form a compact linear program. We then presented a novel graphical platform (CSG) which allows for the quick generation of all relevant non symmetrical clusters. In effect, the CSG serves as a bounded-width window onto the automorphism group of the graphical model. Thus, CSGs may be used to apply tighter approximations within various frameworks of lifted inference.

Acknowledgements

We thank the anonymous reviewers for their comments and useful suggestions. We would also like to thank Eden Chlamtac for fruitful discussions, and for the initial insight

that helped to establish the direction pursued in this work. The authors were supported in part by ISF grant 933/13, the Lynn and William Frankel Center for Computer Science, and by the German-Israeli Foundation (GIF) for Scientific Research and Development, 1180-218.6/2011.

References

Ahmadi, B.; Kersting, K.; Mladenov, M.; and Natarajan, S. 2013. Exploiting symmetries for scaling loopy belief propagation and relational training. *Machine Learning Journal* 92:91–132.

Apsel, U., and Brafman, R. I. 2011. Extended lifted inference with joint formulas. In Cozman, F. G., and Pfeffer, A., eds., *UAI*, 11–18. AUAI Press.

Apsel, U., and Brafman, R. I. 2012. Exploiting uniform assignments in first-order MPE. In de Freitas, N., and Murphy, K. P., eds., *UAI*, 74–83. AUAI Press.

Bödi, R.; Herr, K.; and Joswig, M. 2013. Algorithms for highly symmetric linear and integer programs. *Math. Program.* 137(1-2):65–90.

Bui, H.; Huynh, T.; and Riedel, S. 2013. Automorphism groups of graphical models and lifted variational inference. In *Proceedings of the Twenty-Ninth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-13)*, 132–141. Corvallis, Oregon: AUAI Press.

de Salvo Braz, R.; Amir, E.; and Roth, D. 2005. Lifted first-order probabilistic inference. In Kaelbling, L. P., and Saffiotti, A., eds., *IJCAI*, 1319–1325. Professional Book Center.

Gatermann, K., and Parrilo, P. A. 2004. Symmetry groups, semidefinite programs, and sums of squares. *Journal of Pure and Applied Algebra* 192(1):95–128.

Jha, A.; Gogate, V.; Meliou, A.; and Suciu, D. 2010. Lifted inference from the other side: The tractable features. In *Proc. of the 23rd Annual Conference on Advances in Neural Information Processing Systems (NIPS-2010),*, 973–981.

Karmarkar, N. 1984. A new polynomial-time algorithm for linear programming. *Combinatorica* 4(4):373–396.

Kersting, K.; Ahmadi, B.; and Natarajan, S. 2009. Counting Belief Propagation. In *Proc. of the 25th Conf. on Uncertainty in Artificial Intelligence (UAI–09)*.

Kersting, K. 2012. Lifted probabilistic inference. In *Proceedings of ECAI-2012*. IOS Press.

Khachiyan, L. G. 1979. A polynomial algorithm in linear programming. *Doklady Akademii Nauk SSSR* 244:1093–1096.

Lasserre, J. B. 2002. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM Journal on Optimization* 12(3):756–769.

Lovász, L., and Schrijver, A. 1991. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization* 1(2):166–190.

Margot, F. 2010. Symmetry in integer linear programming. In *50 Years of Integer Programming 1958-2008*. Springer. 647–686.

McKay, B. D., and Piperno, A. 2014. Practical graph isomorphism, {II}. *Journal of Symbolic Computation* 60(0):94 – 112.

Mladenov, M.; Ahmadi, B.; and Kersting, K. 2012. Lifted linear programming. In *15th Int. Conf. on Artificial Intelligence and Statistics (AISTATS 2012)*, 788–797. Volume 22 of JMLR: W&CP 22.

Mladenov, M.; Globerson, A.; and Kersting, K. 2014. Efficient lifting of MAP LP relaxations using k-locality. In *17th Int. Conf. on Artificial Intelligence and Statistics (AISTATS 2014)*.

Niepert, M. 2012. Markov chains on orbits of permutation groups. In *Proc. of the 28th Conf. on Uncertainty in Artificial Intelligence (UAI)*.

Noessner, J.; Niepert, M.; and Stuckenschmidt, H. 2013. Rockit: Exploiting parallelism and symmetry for MAP inference in statistical relational models. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI)*.

Ostrowski, J. 2012. Using symmetry to optimize over the Sherali-Adams relaxation. In *ISCO*, 59–70.

Poole, D. 2003. First-order probabilistic inference. In Gottlob, G., and Walsh, T., eds., *IJCAI*, 985–991. Morgan Kaufmann.

Ravikumar, P.; Agarwal, A.; and Wainwright, M. J. 2010. Message-passing for graph-structured linear programs: Proximal methods and rounding schemes. *The Journal of Machine Learning Research* 11:1043–1080.

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

Schrijver, A. 1998. *Theory of Linear and Integer Programming*. Wiley Series in Discrete Mathematics & Optimization. John Wiley & Sons.

Sherali, H. D., and Adams, W. P. 1990. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.* 3(3):411–430.

Shimony, S. E. 1994. Finding MAPs for belief networks is NP-hard. *Artif. Intell.* 68(2):399–410.

Singla, P., and Domingos, P. 2008. Lifted First-Order Belief Propagation. In *Proc. of the 23rd AAAI Conf. on Artificial Intelligence (AAAI-08)*, 1094–1099.

Sontag, D., and Jaakkola, T. 2007. New outer bounds on the marginal polytope. In *NIPS*.

Sontag, D.; Globerson, A.; and Jaakkola, T. 2011. Introduction to dual decomposition for inference. *Optimization for Machine Learning* 1.

Sontag, D. A. 2010. *Approximate inference in graphical models using LP relaxations*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Van den Broeck, G.; Taghipour, N.; Meert, W.; Davis, J.; and Raedt, L. D. 2011. Lifted probabilistic inference by first-order knowledge compilation. In *IJCAI*, 2178–2185.

Yedidia, J. S.; Freeman, W. T.; and Weiss, Y. 2003. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium* 8:236–239.