Lifted Probabilistic Inference: An MCMC Perspective

Mathias Niepert
Universität Mannheim
mniepert@gmail.com

Abstract

The general consensus seems to be that lifted inference is concerned with exploiting model symmetries and grouping indistinguishable objects at inference time. Since first-order probabilistic formalisms are essentially template languages providing a more compact representation of a corresponding ground model, lifted inference tends to work especially well in these models. We show that the notion of indistinguishability manifests itself on several different levels – the level of constants, the level of ground atoms (variables), the level of formulas (features), and the level of assignments (possible worlds). We discuss existing work in the MCMC literature on exploiting symmetries on the level of variable assignments and relate it to novel results in lifted MCMC.

1 Introduction

Numerous algorithms exploit model symmetries with the goal of reducing the complexity of the computational problems at hand. A considerable amount of attention to approaches utilizing model symmetries has been given by researchers working on "lifted probabilistic inference [18]." Lifted inference is mainly motivated by the large probabilistic graphical models resulting from statistical relational formalism such as Markov logic networks [20]. The unifying theme of lifted probabilistic inference is that inference on the level of instantiated formulas is avoided and instead lifted to the first-order level. Notable approaches are algorithms for lifted belief propagation [23, 10], bi-simulation-based approximate inference algorithms [22], and first-order knowledge compilation techniques [24, 8], to name but a few.

We recently proposed the use of permutation groups and, more generally, group theoretical concepts and algorithms to represent and manipulate symmetries in probabilistic models [15]. The representation of symmetries with irredundant generators of permutations groups provides an exponential compression, that is, a representation logarithmic in the number of permutations. Since first-order models often exhibit strong topological symmetries, permutation groups offer a compact and well-understood representation. Moreover, numerous efficient group theoretical algorithms are implemented in comprehensive open-source group algebra frameworks such as Gap[7]. In recent work, we described the construction of colored undirected graphs whose automorphism groups are equivalent to those of the probabilistic graphical models under consideration. Moreover, we have shown that existing algorithms such as Saucy[4] and Nauty[12] compute automorphism groups of the resulting colored graphs very efficiently even for models involving millions of variables [15].

Symmetries on different syntactical levels of statistical relational formalism ultimately lead to symmetries in the space of joint variable assignments. This space of possible assignments corresponds to the state space of Monte Carlo Markov chains such as the Gibbs sampler that are often used for approximate probabilistic inference. Since the permutation group modeling the symmetries induces a partition (the so-called orbit partition) on the state space of these Markov chains, we investigate whether this can be exploited for more efficient MCMC approaches to probabilistic inference. The basic idea is that the *lifted* Markov chains[1] implicitly or explicitly operate on the partition of the state space instead of the space of individual assignments. We also describe orbital Markov chains, a recent contribution to lifted MCMC [15]. An orbital Markov chain is always derived from an existing Markov chain

[1] Please note that in the standard MCMC literature a *lifting* of a Markov chain [3] is *not* the same as our notion of a lifted Markov chain. We will later come back to this.

so as to leverage the symmetries in the underlying model. Under mild conditions, orbital Markov chains have the same convergence properties as chains operating on the state space partition without the need to explicitly compute this partition.

We conducted several experiments verifying that orbital Markov chains converge faster to the true distribution than state of the art Markov chains on well-motivated and established sampling problems such as the problem of sampling independent sets from graphs and the problem of computing the marginal single variable probabilities of large Markov logic networks.

2 Background

We first recall basic concepts of group theory and finite Markov chains both of which are crucial for understanding the presented work. Please note that these should not be difficult to graps even for readers unfamiliar with group theory.

2.1 Group Theory

A symmetry of a discrete object is a structure-preserving bijection on its components. A natural way to represent symmetries are permutation groups. A permutation group $\mathfrak{G}$ *acting on* a finite set Ω is a finite set of bijections $\mathfrak{g} : \Omega \rightarrow \Omega$ that form a group. Let Ω be a finite set and let $\mathfrak{G}$ be a permutation group acting on Ω. If $\alpha \in \Omega$ and $\mathfrak{g} \in \mathfrak{G}$ we write $\alpha^{\mathfrak{g}}$ to denote the image of α under $\mathfrak{g}$. A cycle $(\alpha_1\ \alpha_2\ ...\ \alpha_n)$ represents the permutation that maps α_1 to α_2, α_2 to α_3,..., and α_n to α_1. Every permutation can be written as a product of disjoint cycles where each element that does not occur in a cycle is understood as being mapped to itself. We define a relation $\sim$ on Ω with $\alpha \sim \beta$ if and only if there is a permutation $\mathfrak{g} \in \mathfrak{G}$ such that $\alpha^{\mathfrak{g}} = \beta$. The relation partitions Ω into equivalence classes which we call *orbits*. We use the notation $\alpha^{\mathfrak{G}}$ to denote the orbit $\{\alpha^{\mathfrak{g}} \mid \mathfrak{g} \in \mathfrak{G}\}$ containing α. Let $f : \Omega \rightarrow \mathbb{R}$ be a function from Ω into the real numbers and let $\mathfrak{G}$ be a permutation group acting on Ω. We say that $\mathfrak{G}$ is an *automorphism group for* (Ω, f) if and only if for all $\omega \in \Omega$ and all $\mathfrak{g} \in \mathfrak{G}$, $f(\omega) = f(\omega^{\mathfrak{g}})$.

2.2 Finite Markov Chains

Given a finite set Ω a *finite Markov chain* defines a random walk $(X_0, X_1, ...)$ on elements of Ω with the property that the conditional distribution of X_{n+1} given $(X_0, X_1, ..., X_n)$ depends only on X_n. For all $x, y \in \Omega$ $P(x, y)$ is the chain's probability to transition from x to y, and $P^t(x, y) = P^t_x(y)$ the probability of being in state y after t steps if the chain starts at x. A Markov chain is *irreducible* if for all $x, y \in \Omega$ there exists a t such that $P^t(x, y) > 0$ and *aperiodic* if for all $x \in \Omega$, $\mathsf{gcd}\{t \geq 1 \mid P^t(x, x) > 0\} = 1$. A chain that is both irreducible and aperiodic is called an *ergodic* chain and converges to a unique stationary distribution.

The total variation distance d_{tv} of the Markov chain from its stationary distribution π at time t with initial state x is defined by

$$d_{\mathsf{tv}}(P^t_x, \pi) = \frac{1}{2} \sum_{y \in \Omega} |P^t(x, y) - \pi(y)|.$$

For $\varepsilon > 0$, let $\tau_x(\varepsilon)$ denote the least value T such that $d_{\mathsf{tv}}(P^t_x, \pi) \leq \varepsilon$ for all $t \geq T$. The *mixing time* $\tau(\varepsilon)$ is defined by $\tau(\varepsilon) = \max\{\tau_x(\varepsilon) \mid x \in \Omega\}$. We say that a Markov chain is *rapidly mixing* if the mixing time is bounded by a polynomial in n and $\log(\varepsilon^{-1})$, where n is the size of each configuration in Ω.

3 Symmetry in Statistical Relational Artificial Intelligence

The notion of lifted probabilistic inference was first introduced in the context of first-order variable elimination, a variation of variable elimination, taking advantage of the symmetries in statistical relational models [18]. Following Poole's work, several algorithms for lifted probabilistic inference were developed such as lifted and counting belief propagation [23, 10], bi-simulation based approximate inference [22], general purpose MCMC algorithm for relational models [13] and, more recently, first-order knowledge compilation techniques [24, 8]. Naturally, there is a close connection between the concept of symmetry and lifted inference. For instance, lifted belief propagation identifies and clusters indistinguishable ground atoms and features by keeping track of the messages send and received by each of the corresponding nodes in a factor graph [23, 10]. Bi-simulation type procedures group indistinguishable elements and, therefore, exploit symmetry in the model as well [22].

We have proposed the use of group theory and, in particular, permutation groups to compactly represent symmetries in graphical models [15]. There are several reasons to consider group theory and permutation groups a natural representation of symmetries in graphical models. First, an irredundant set of generators of a permutation group ensures exponential compression. For instance, for a set of n exchangeable binary random variables, the permutation group acting on the variables is the symmetric group on n which has $n!$ permutations. However, we only need at most $n-1$ irredundant generators to represent this permutation group. In addition to the compact representation, group theory also provides numerous remarkably efficient algorithms for manipulating and sampling from

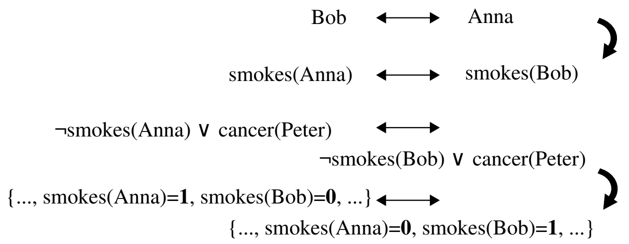
Bob
Anna
smokes(Anna)
smokes(Bob)
¬smokes(Anna) ∨ cancer(Peter)
¬smokes(Bob) ∨ cancer(Peter)
{..., smokes(Anna)=**1**, smokes(Bob)=**0**, ...}
{..., smokes(Anna)=**0**, smokes(Bob)=**1**, ...}

Figure 1: Symmetry in the model is observable on different syntactical levels of the relational model. The level of constants, the level of ground atoms (variables), the level of clauses (features) and the level of possible worlds (assignments). Each permutation group acting on the set of constants induces a permutation group acting on the set of ground atoms. The latter induces a permutation group acting on the set of features. This permutation partitions (a) the variables and feature and (b) the assignment space.

groups. The product replacement algorithm [2], for instance, samples group elements uniformly at random with impressive performance.

Symmetry in statistical relational languages manifests itself at various syntactic levels ranging from the set of constants to the assignment space. There is often symmetry at the level of constants. In the well-known social network model [23] without evidence, for example, we have that the constants are indistinguishable meaning that swapping two constants leads to an isomorphic statistical relational model. Now, the permutations on the constant level induce permutations on the level of ground atoms and formulas. From the irredundant generators of the permutation group modeling the symmetries on the constant level we can directly compute the irredundant generators of the permutation group modeling the corresponding symmetries on the ground level. Indeed, it is well-known that isomorphisms between permutation groups always map irredundant generators in one group to irredundant generators in the other. However, symmetry on the ground level does not necessarily lead to symmetry on the constant level. Similarily, while symmetry on the ground level induces symmetry on the space of assignments to the random variables this is not true for the other direction. Figure 1 depicts the different syntactical levels on which symmetries can arise.

Most existing lifted inference algorithms *implicitly* exploit symmetry on the ground level. Examples are lifted variable elimination [18] as well as counting belief propagation [23, 10] approaches that cluster variables and features whose factor nodes would send and receive the same messages were belief propagation run on the ground model. A recently developed approach maps weighted formulas to colored undirected graphs and applies graph automorphism algorithms to compute the symmetries of the log-linear models defined over the weighted formulas [15]. The resulting permutation groups partition the (exponential) space of variable assignments when acting on it. Since the state space of MCMC approaches is identical to the assignment space of the probabilistic graphical models, we will investigate whether and to what extend the partition induced by the models' symmetries can be leveraged for more efficient MCMC algorithms.

4 Lifted MCMC

We have seen that symmetries on different syntactical levels of statistical relational formalism ultimately lead to symmetries in the space of joint variable assignments. Now, the space of possible variable assignments is the state space of Monte Carlo Markov chains such as the Gibbs sampler that are often used for approximate probabilistic inference. Since the permutation group modeling the symmetries induces a partition (the so-called orbit partition) on the state space of these Markov chains, we will investigate whether this can be exploited for more efficient MCMC approaches to probabilistic inference. The basic idea is that the *lifted* MCMC algorithms implicitly or explicitly operate on the partition of the state space instead of the original state space.

4.1 Lumping

A *lumping* (also: collapsing, projection) of a Markov chains is a compression of its state space which is possible under certain conditions on the transition probabilities of the original Markov chain [1, 5]. The following definition formalizes the notion.

Definition 4.1. Let $\mathcal{M}$ be a Markov chain with transition matrix P and state space Ω, and let $\mathcal{C} = \{C_1, ..., C_n\}$ be a partition of the state space. If for all $C_i, C_j \in \mathcal{C}$ and all $j', j'' \in C_j$

$$\sum_{i' \in C_i} P(i', j') = \sum_{i' \in C_i} P(i', j'')$$

then $\mathcal{M}$ is ordinary lumpable. If, in addition, $\pi(j') = \pi(j'')$ for all $j', j'' \in C_j$ and all $C_j \in \mathcal{C}$ then $\mathcal{M}$ is exactly lumpable.

Let $\hat{\pi}$ be the stationary distribution of the *quotient Markov chain*, that is, the exactly lumped Markov chain whose state space is $\mathcal{C}$. Then, the probability $\pi(i)$ of a state $i \in C_i \subseteq \Omega$ of the original chain can be computed as $\pi(i) = \hat{\pi}(i)/|C_i|$.

The benefit of lumping a Markov chain is the potentially much smaller state space and ultimately more rapid mixing. For instance, consider the case of n binary random variables that are exchangeable[2]. Here, the natural choice of a partition of the state space is $\{C_0, C_1, ..., C_n\}$ where each C_i contains the states with Hamming weight i, that is, the states with i non-zeros. Please note that the C_i's are the orbits (equivalence classes) of the orbit partition of the permutation group acting on the set of states (variable assignments). Instead of 2^n states the resulting lumped Markov chain has only $n + 1$ states and mixes more rapidly than the original one. Figure 2 depicts (a) a fragment of a finite Markov chain with non-zero probability transitions indicated by arrows and (b) a lumped Markov chain that bundles several states of the original chain into a single one of the lumped chain.

The crucial question is whether the explicit construction of the lumped chain is computationally feasible. After all, if the computation of lumped Markov chains was intractable we would not have gained much. Unfortunately, it turns out that the explicit construction of the lumped state space is indeed intractable. Computing the coarsest lumping quotient of a Markov chain with a bi-simulation procedure is linear in the *number of non-zero probability transitions* of the chain [5] and, hence, in most cases exponential in the number of random variables. Moreover, other theoretical results show that special cases of the lumping problem are also intractable. The results are negative even for the important special case of partitions resulting from permutation groups acting on the state space of the Markov chains. It is known that, given a permutation group acting on the state space, merely computing the number of equivalence classes of the resulting orbit partition of the state space is a #P-complete problem [9].

We hypothesize that the intractability of the explicit construction of the lumped chain's state space is the main reason that the technique of lumping, while well-understood on a theoretical level, has not been seriously considered by communities that apply Markov chain Monte Carlo methods to large-scale applications requiring probabilistic inference. We are not aware of MCMC approaches to probabilistic reasoning that leverage the theory of lumping.

4.2 Orbital Markov Chains

A recent study of Markov chains that exploit state space symmetries has shown that, under certain circumstances, the *explicit* computation of the partition of the state space is not necessary to achieve the same computational gains as the lumped chain [15]. The basic idea is that we only need, for each $\omega \in \Omega$, an efficient way to sample uniformly at random from $[\omega]$ the equivalence class containing ω. It was shown that the product replacement algorithm [2] provides such an efficient method of sampling uniformly from the equivalence classes induced by a permutation group. The novel family of Markov chains was termed *orbital Markov chains* [15]. An orbital Markov chain is always derived from an existing Markov chain so as to leverage the symmetries in the underlying model. In the presence of symmetries orbital Markov chains are able to perform wide-ranging transitions reducing the time until convergence. In the absence of symmetries they are equivalent to the original Markov chains. Orbital Markov chains only require a generating set of a permutation group $\mathfrak{G}$ acting on the chain's state space as additional input. As we have previously discussed, these sets of generators are computable with algorithms that derive permutation groups for colored undirected graphs such as Saucy and Nauty or by means of other algorithms that have been developed in the lifted inference literature.

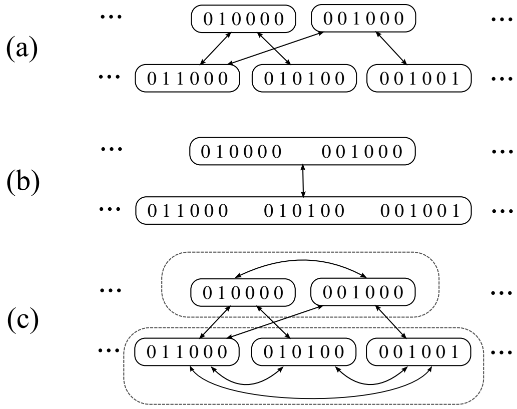

Figure 2: (a) A fragment of a finite state space of a Markov chain with non-zero transition probabilities indicated by directed arcs. (b) A lumping of the state space. Instead of moving between individual states, the lumped chain moves between classes of states of the original chain. (c) The benefits of lumping are also achievable by sampling uniformly at random from the implicit equivalence classes (orbits) in each step.

Let Ω be a finite set, let $\mathcal{M}' = (X'_0, X'_1, ...)$ be a Markov chain with state space Ω, let π be a stationary distribution of $\mathcal{M}'$, and let $\mathfrak{G}$ be an automorphism group for (Ω, π). The *orbital Markov chain* $\mathcal{M} = (X_0, X_1, ...)$ *for* $\mathcal{M}'$ is a Markov chain which at each integer time $t+1$ performs the following steps:

1. Let X'_{t+1} be the state of the *original* Markov chain $\mathcal{M}'$ at time $t + 1$;

[2] In SRL models this is equivalent to the case of all ground atoms being indistinguishable.

2. Sample X_{t+1}, the state of the orbital Markov chain $\mathcal{M}$ at time $t+1$, uniformly at random from ${X'}_{t+1}^{\mathfrak{G}}$, the orbit of X'_{t+1}.

The orbital Markov chain $\mathcal{M}$, therefore, runs at every time step $t \geq 1$ the original chain $\mathcal{M}'$ first and samples the state of $\mathcal{M}$ at time t uniformly at random from the orbit of the state of the original chain $\mathcal{M}'$ at time t. Figure 2 (c) depicts a fragment of the orbital Markov chain for the original Markov chain (a). Instead of computing the equivalence of the state space explicitly (b) novel transitions are introduced that make the chain behave *as if it was lumped*.

Given a state X_t and a permutation group $\mathfrak{G}$ orbital Markov chains sample an element from ${X_t}^{\mathfrak{G}}$, the orbit of X_t, uniformly at random. By the orbit-stabilizer theorem this is equivalent to sampling an element $\mathfrak{g} \in \mathfrak{G}$ uniformly at random and computing ${X_t}^{\mathfrak{g}}$. Sampling group elements uniformly at random is a well-researched problem [2] and computable in polynomial time in the size of the generating sets with product replacement algorithms [17]. These algorithms are implemented in several group algebra systems such as GAP[7] and exhibit remarkable performance. Once initialized, product replacement algorithms can generate pseudo-random elements by performing, depending on the variant, 1 to 3 group multiplications. We could verify that the overhead of step 2 during the sampling process is indeed negligible.

The following theorem relates properties of the orbital Markov chain to those of the Markov chain it is derived from. A detailed proof can be found in the appendix.

Theorem 4.2 (Niepert [15])**.** *Let Ω be a finite set and let $\mathcal{M}'$ be a Markov chain with state space Ω and transition matrix P'. Moreover, let π be a probability distribution on Ω, let $\mathfrak{G}$ be an automorphism group for (Ω, π), and let $\mathcal{M}$ be the orbital Markov chain for $\mathcal{M}'$. Then,*

(a) if $\mathcal{M}'$ is aperiodic then $\mathcal{M}$ is also aperiodic;

(b) if $\mathcal{M}'$ is irreducible then $\mathcal{M}$ is also irreducible;

(c) if π is a reversible distribution for $\mathcal{M}'$ and, for all $\mathfrak{g} \in \mathfrak{G}$ and all $x, y \in \Omega$ we have that $P'(x,y) = P'(x^{\mathfrak{g}}, y^{\mathfrak{g}})$, then π is also a reversible and, hence, a stationary distribution for $\mathcal{M}$.

The condition in statement (c) requiring for all $\mathfrak{g} \in \mathfrak{G}$ and all $x, y \in \Omega$ that $P'(x,y) = P'(x^{\mathfrak{g}}, y^{\mathfrak{g}})$ expresses that the original Markov chain is compatible with the symmetries captured by the permutation group $\mathfrak{G}$. This weak assumption is met by all of the practical Markov chains we are aware of and, in particular, Metropolis chains and Gibbs sampler.

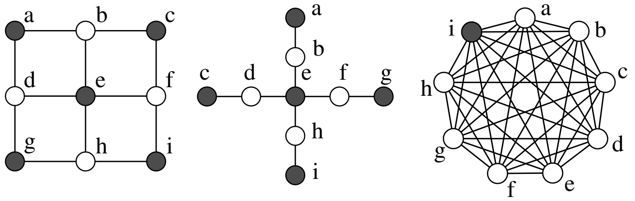

Figure 3: From left to right: the 3-*grid*, the 3-*connected cliques*, and the 3-*complete graph* models.

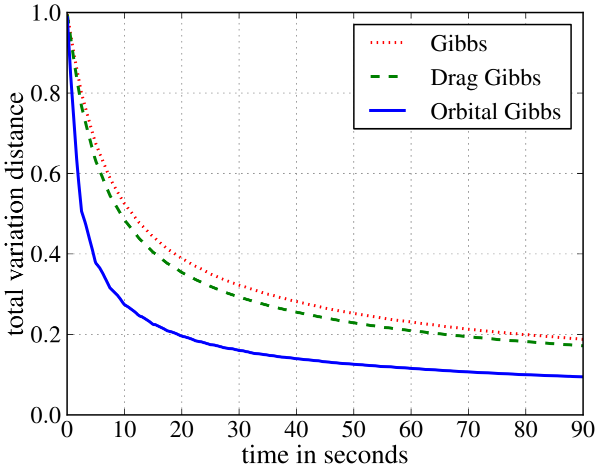

Figure 4: The results of the three Gibbs samplers for the 5-*grid model*.

5 Experiments

We conducted experiments on two different well-established classes of graphical models. First, we selected the *insert/delete* Markov chain for independent sets of graphs for the experiments. Sampling independent sets is a classical problem motivated by numerous applications and with a considerable amount of recent research devoted to it [11, 6].

Let $G = (V, E)$ be a graph. A subset X of V is an *independent set* if $\{v, w\} \notin E$ for all $v, w \in X$. Let $\mathcal{I}(G)$ be the set of all independent sets in a given graph G and let λ be a positive real number. The partition function $Z = Z(\lambda)$ and the corresponding probability measure π_λ on $\mathcal{I}(G)$ are defined by

$$Z = Z(\lambda) = \sum_{X \in \mathcal{I}(G)} \lambda^{|X|} \quad \text{and} \quad \pi_\lambda(X) = \frac{\lambda^{|X|}}{Z}.$$

Approximating the partition function and sampling from $\mathcal{I}(G)$ can be accomplished using a rapidly mixing Markov chain with state space $\mathcal{I}(G)$ and stationary distribution π_λ. The simplest Markov chain for independent sets is the so-called *insert/delete* chain [6]. If X_t is the state at time t then the state at time $t+1$ is determined by the following procedure:

1. Select a vertex $v \in V$ uniformly at random;
2. If $v \in X_t$ then let $X_{t+1} = X_t \backslash \{v\}$ with probability $1/(1+\lambda)$;
3. If $v \notin X_t$ and v has no neighbors in X_t then let $X_{t+1} = X_t \cup \{v\}$ with probability $\lambda/(1+\lambda)$;
4. Otherwise let $X_{t+1} = X_t$.

Using a path coupling argument one can show that the *insert/delete* chain is rapidly mixing for $\lambda \leq 1/(\Delta - 1)$ where Δ is the maximum degree of the graph [6]. We can turn the *insert/delete* Markov chain into the orbital *insert/delete* Markov chain $\mathcal{M}(\mathcal{I}(G))$ simply by adding the following fifth step:

5. Sample X_{t+1} uniformly at random from its orbit.

As a corollary to Theorem 4.2, we have that the orbital *insert/delete* chain for independent sets is aperiodic, irreducible, and has π_λ as its unique stationary distribution [15].

We compared the performance of the orbital Markov chains for independent sets of graphs with state-of-the-art algorithms for sampling independent sets [6]. We used GAP, a system for computational discrete algebra, and the ORB package[3] to implement the sampling algorithms. The experiments can easily be replicated by installing GAP and the ORB package and by running the GAP files available at a dedicated code repository[4]. For the evaluation of the sampling algorithms we selected three different graph topologies exhibiting varying degrees of symmetry:

The *k-grid model* is the 2-dimensional $k \times k$ grid. An instance of the model for $k = 3$ is depicted in Figure 3 (left). Here, the generating set of the permutation group $\mathfrak{G}$ computed by SAUCY is $\{(a\ c)(d\ f)(g\ i), (a\ i)(b\ f)(d\ h)\}$ and $|\mathfrak{G}| = 8$.
The *k-connected cliques model* is a graph with $k+1$ distinct cliques each of size $k-1$ and each connected with one edge to the same vertex. Statistical relational formalisms such as Markov logic networks often lead to similar graph topologies. An instance for $k = 3$ is depicted in Figure 3 (center). Here, the generating set of $\mathfrak{G}$ computed by SAUCY is $\{(a\ g)(b\ f), (a\ c)(b\ d), (a\ i)(b\ h)\}$ and $|\mathfrak{G}| = 24$.
The *k-complete graph model* is a complete graph with k^2 vertices. Figure 3 (right) depicts an instance for $k = 3$. Here, the generating set of $\mathfrak{G}$ computed by SAUCY is $\{(b\ c), (b\ d), (b\ e), (b\ f), (b\ g), (b\ h), (b\ i), (a\ b)\}$ and $|\mathfrak{G}| = 9! = 362880$.

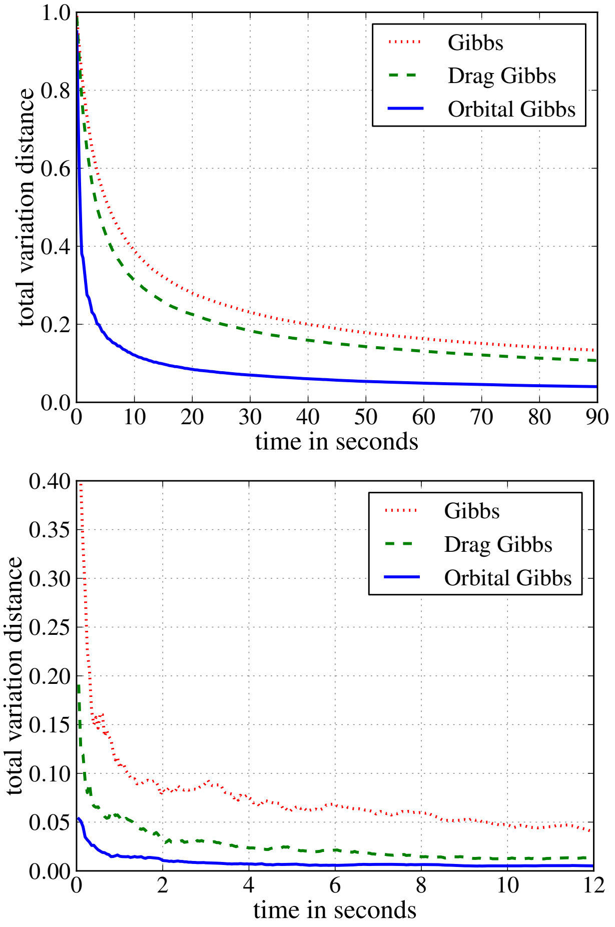

Figure 5: The results of the three Gibbs samplers for the 5-*connected cliques* (top) and the 5-*complete graph* (bottom) model.

SAUCY needed about 5 ms to compute the sets of generators for the permutation groups of the three models. We generated samples of the probability measure π_λ on $\mathcal{I}(G)$ for $\lambda = 1$ and the three graph topologies by running (a) the *insert/delete* chain, (b) the *insert/delete/drag* chain [6], and (c) the orbital *insert/delete* chain. Each chain was started in the state corresponding to the empty set and no burn-in period was used. The orbital *insert/delete* chain did not require more RAM and needed 50 microseconds per sample which amounts to an overhead of about 25% relative to the 40 microseconds of the *insert/delete* chain. The 25% overhead remained constant and independent of the size of the graphs. Since the sampling algorithms create large files with all accumulated samples, I/O overhead is included in these times. For each of the three topologies and each of the three Gibbs

[3] http://www.gap-system.org/Packages/orb.html
[4] http://code.google.com/p/lifted-mcmc/

samplers, we computed the total variation distance between the distribution approximated using *all* accumulated samples and the true distribution π_1. Figure 4 plots the total variation distance over elapsed time for the *k-grid model* for $k = 5$. The orbital *insert/delete* chain (Orbital Gibbs) converges the fastest. The *insert/delete/drag* chain (Drag Gibbs) converges faster than the *insert/delete* chain (Gibbs). The same results are observable for the other graph topologies (see Figures 5) where the orbital Markov chain outperforms the others. The larger the cardinalities of the orbits induced by the symmetries the faster converges the orbital Gibbs sampler relative to the other chains.

Moreover, we conducted experiments with the well-established social network Markov logic network (the smokes-cancer MLN) exactly as specified in [23]. Here we created two ground MLNs with 50 and 100, respectively, people in the domain, leading to Markov networks with 2600 and 10200 variables, respectively. Building the ground models took only a fraction of a second. We proceeded to apply the symmetry detection algorithm [15] taking 24 and 136 ms, respectively, to compute the irredundant generators of the automorphism group of the models. For n people in the domain, there are $n - 1$ irredundant generators of the automorphism group and the group has size $n!$ which is exactly the size of the symmetric group on n. Please note that, based on our observation of indistinguishability of objects on different syntactical levels of the model, it is actually not necessary to use symmetry detection algorithms in this case. The irredundant generators of the symmetric group representing the symmetries on the level of constants can be directly used to compute the irredundant generators for the permutation group representing the symmetries on the level of ground atoms and formulas.

Finally, we compared the standard Gibbs sampler, Alchemy's MC-SAT algorithm [19], and the orbital Gibbs sampler on the models. The overhead of the product replacement algorithm was again negligible and far outweighed by the faster convergence of the orbital chain. Figure 5 plots the symmetric Kullback-Leibler divergence for the single variable marginals.

6 Discussion

We have presented a novel perspective on lifted inference. Instead of directly operating on the space of joint variable assignments, *orbital* Markov chains operate on a symmetry-induced partition of this space. We related lifted MCMC to the notion of lumping of Markov chains. Instead of computing the partition of the state space explicitly which is usually intractable, orbital Markov chains operate on the original state space while having convergence properties identical to the corresponding lumped Markov chain. We want to point out that in the MCMC literature a lifting of a Markov chain [3] is *not* the same as what has been coined lifted inference by the statistical relational AI community. Quite the opposite, instead of operating on a more compact state space, lifting in the classical sense introduces additional states. Nevertheless, there might be interesting relationships between lumping, lifting and lifted inference.

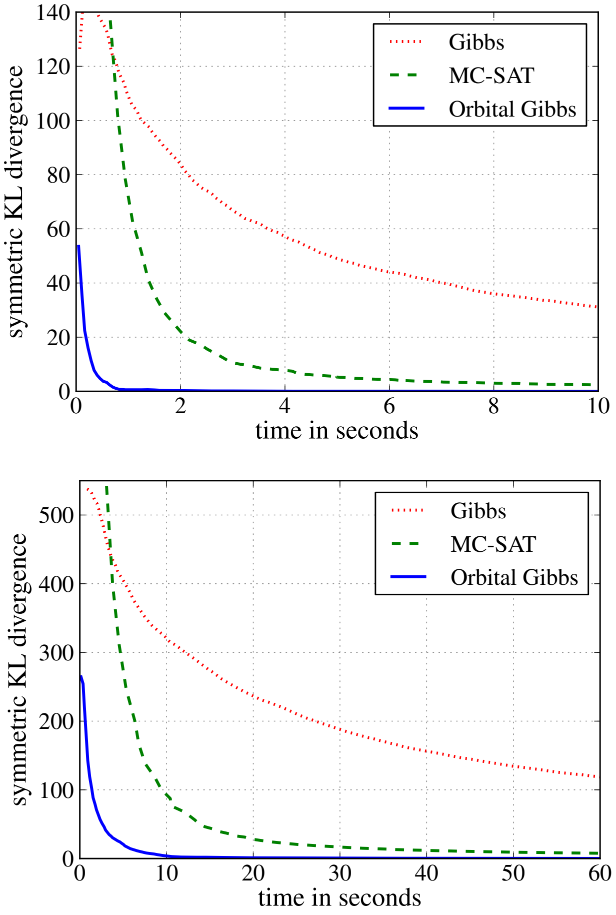

Figure 6: The results of the standard Gibbs sampler, Alchemy's MC-SAT algorithm, and the orbital Gibbs sampler for the social network MLN with 50 (top) and 100 (bottom) people in the domain.

Future work will include the integration of orbital Markov chains with algorithms for marginal as well as maximum a-posteriori inference. We will also apply the symmetry detection approach to make existing inference algorithms more efficient by, for instance, using symmetry breaking constraints in combinatorial optimization approaches to maximum a-posteriori inference in Markov logic networks (cf. [21, 14, 16]).

Acknowledgments

Many thanks to Manfred Jaeger, a conversation with whom sparked the idea of using symmetries on the level of constants to directly determine symmetries on the ground level.

References

[1] P. Buchholz. Exact and ordinary lumpability in finite markov chains. *Journal of Applied Probability*, 31(1):59–75, 1994.

[2] F. Celler, C. R. Leedham-Green, S. H. Murray, A. C. Niemeyer, and E. O'brien. Generating random elements of a finite group. *Communications in Algebra*, 23(13):4931–4948, 1995.

[3] F. Chen, L. Lovász, and I. Pak. Lifting markov chains to speed up mixing. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, STOC '99, pages 275–281, 1999.

[4] P. T. Darga, K. A. Sakallah, and I. L. Markov. Faster symmetry discovery using sparsity of symmetries. In *Proceedings of the 45th annual Design Automation Conference*, pages 149–154, 2008.

[5] S. Derisavi, H. Hermanns, and W. H. Sanders. Optimal state-space lumping in markov chains. *Inf. Process. Lett.*, 87(6):309–315, 2003.

[6] M. Dyer and C. Greenhill. On markov chains for independent sets. *Journal of Algorithms*, 35(1):17–49, 2000.

[7] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.4.12*, 2008.

[8] V. Gogate and P. Domingos. Probabilistic theorem proving. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, pages 256–265, 2011.

[9] L. A. Goldberg. Computation in permutation groups: counting and randomly sampling orbits. In *Surveys in Combinatorics*, pages 109–143. Cambridge University Press, 2001.

[10] K. Kersting, B. Ahmadi, and S. Natarajan. Counting belief propagation. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 277–284, 2009.

[11] M. Luby and E. Vigoda. Fast convergence of the glauber dynamics for sampling independent sets. *Random Struct. Algorithms*, 15(3-4):229–241, 1999.

[12] B. D. McKay. Practical graph isomorphism. *Congressus Numerantium*, 30:45–87, 1981.

[13] B. Milch and S. J. Russell. General-purpose mcmc inference over relational structures. In *Proceedings of the 22nd Conference in Uncertainty in Artificial Intelligence*, pages 349–358, 2006.

[14] M. Niepert. A Delayed Column Generation Strategy for Exact k-Bounded MAP Inference in Markov Logic Networks. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 2010.

[15] M. Niepert. Markov chains on orbits of permutation groups. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence*, 2012.

[16] M. Niepert, J. Noessner, and H. Stuckenschmidt. Log-linear description logics. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 2153–2158, 2011.

[17] I. Pak. The product replacement algorithm is polynomial. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 476–485, 2000.

[18] D. Poole. First-order probabilistic inference. In *Proceedings of the 18th Joint Conference on Artificial Intelligence*, pages 985–991, 2003.

[19] H. Poon and P. Domingos. Sound and efficient inference with probabilistic and deterministic dependencies. In *Proceedings of the 21st Conference on Artificial Intelligence*, pages 458–463, 2006.

[20] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2), 2006.

[21] S. Riedel. Improving the accuracy and efficiency of map inference for markov logic. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 468–475, 2008.

[22] P. Sen, A. Deshpande, and L. Getoor. Bisimulation-based approximate lifted inference. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 2009.

[23] P. Singla and P. Domingos. Lifted first-order belief propagation. In *Proceedings of the 23rd Conference on Artificial Intelligence*, pages 1094–1099, 2008.

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