ECAI 2012
Luc De Raedt et al. (Eds.)

doi:10.3233/978-1-61499-098-7-33

Lifted Probabilistic Inference

Kristian Kersting[1]

Abstract. Many AI problems arising in a wide variety of fields such as machine learning, semantic web, network communication, computer vision, and robotics can elegantly be encoded and solved using probabilistic graphical models. Often, however, we are facing inference problems with symmetries and redundancies only implicitly captured in the graph structure and, hence, not exploitable by efficient inference approaches. A prominent example are probabilistic logical models that tackle a long standing goal of AI, namely unifying first-order logic — capturing regularities and symmetries — and probability — capturing uncertainty. Although they often encode large, complex models using few rules only and, hence, symmetries and redundancies abound, inference in them was originally still at the propositional representation level and did not exploit symmetries. This paper is intended to give a (not necessarily complete) overview and invitation to the emerging field of lifted probabilistic inference, inference techniques that exploit these symmetries in graphical models in order to speed up inference, ultimately orders of magnitude.

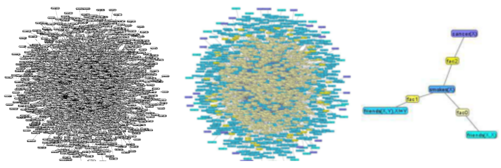

Figure 1. Symmetries in a graphical model can be exploited to speed up inference. (Left) A graphical model with thousands of nodes and factors for a social network inference problem. (Middle) When running e.g. (loopy) belief propagation (BP), many nodes and factors could be indistinguishable (as indicated by the colors) in terms of the BP computations due to shared parameters. (Right) By grouping indistinguishable nodes together, a lifted factor graph exploits these regularities using just four nodes and three factors, producing the same single node marginals but in a fraction of time. (Best viewed in color)

1 Introduction

In the first paragraph of his book, *Symmetry*, Hermann Weyl writes "... symmetric means something like well-proportioned, well-balanced, and symmetry denotes that sort of concordance of several parts by which they integrate into a whole" [69]. Symmetries can be found almost everywhere, in arabesques and French gardens as, in the rose windows and vaults in Gothic cathedrals, in the meter, rhythm, and melody of music, in the metrical and rhyme schemes of poetry as well as in the patterns of steps when dancing. Symmetric faces are even said to be more beautiful to humans. So, symmetry is both a conceptual and a perceptual notion often associated with beauty-related judgments [71]. Or, to quote Hermann Weyl again "Beauty is bound up with symmetry".

This link between symmetry and beauty is often made by scientists. Why is this link so prominent in science? In physics, for instance, symmetry is linked to beauty in that symmetry describes the invariants of nature, which, if discerned could reveal the fundamental, true physical reality [71]. In fact[2], "at the heart of relativity theory, quantum mechanics, string theory, and much of modern cosmology lies one concept: symmetry." In mathematics, as Herr and Bödi note, "we expect objects with many symmetries to be uniform and regular, thus not too complicated" [27]. Therefore, it is not surprising that symmetries have also been explored in many AI tasks such as (mixed–)integer programming [37, 4], SAT and CSP [54, 64] as well as MDPs [17, 50].

Surprisingly, symmetries have not been the subject of interest within probabilistic inference. Only recently the first efforts were made to employ symmetries within probabilistic inference. In 2003, Poole presented in his seminal paper on "First-Order Probabilistic Inference" an algorithm to reason about multiple individuals, where we may know particular facts about some of them, but want to treat the others as a group [47]. This was the starting point of the very active research field called "lifted probabilistic inference". Since then, several inference approaches that exploit symmetries have been proposed, see e.g. [14, 40, 55, 8, 68, 24] among others, and proven successful in many AI tasks and applications such as information retrieval, satisfiability, boolean model counting, semantic role labeling, Kalman filtering, Page Rank, Label Propagation, citation matching, entity resolution, link prediction in social networks, information broadcasting, market analysis, tracking of objects in videos, and biomolecular event prediction. The lifted approaches are often faster, more compact and provide more structure for optimization.

Lifted probabilistic inference is mainly triggered by the recent success of statistical relational learning, see e.g. [21, 13, 12] for overviews, which tackles a long standing goal of AI — namely unifying first-order logic (capturing regularities and symmetries) and and probability (capturing uncertainty) — that can be traced back to Nils Nilsson's seminal paper on "Probabilistic Logic" [45]. Probabilistic logical languages provide powerful formalisms for knowledge representation and inference. They allow one to compactly represent complex relational and uncertain knowledge. For instance, in the friends-and-smokers Markov logic network (MLN) [51], the weighted formula $1.1 : \texttt{fr}(\texttt{X},\texttt{Y}) \Rightarrow (\texttt{sm}(\texttt{X}) \Leftrightarrow \texttt{sm}(\texttt{Y}))$ encodes that friends in a social network tend to have similar smoking habits. Yet performing inference in these languages is extremely costly, especially if it is done at the propositional level. Instantiating all atoms from the formulas in a such a model induces a standard graphical model with symmetric, repeated potential structures for all grounding combinations, see Fig. 1(Left). Lifted probabilistic infer-

[1] Institute of Geodesy and Geoinformation, University of Bonn, Germany, Knowledge Discovery Department, Fraunhofer IAIS, Sankt Augustin, Germany, email: kristian.kersting@iais.fraunhofer.de
[2] The publisher about Ian Stewart's book, *Why Beauty Is Truth* [60].

ence approaches have rendered many of these large, previously intractable problems quickly solvable by exploiting the induced redundancies. As a sneak preview, lifted (loopy) belief propagation (BP) approaches [58, 30, 16, 31, 25, 3] intuitively automatically group variables and factors of a graphical model together if they have identical computation trees (i.e., the tree-structured "unrolling" of the graphical model computations rooted at the nodes) as indicated by the colors in Fig. 1(Middle,Right). Then, they run a modified BP on this lifted, often orders of magnitude smaller network.

This paper is intended to give a (not necessarily complete) overview and invitation to the emerging field of lifted probabilistic inference. Laying bare the ideas will hopefully inspire others to join us in exploring the frontiers and the yet unexplored areas.

We proceed as follows. First, we illustrate symmetries in graphical models without referring to any specific inference algorithm. Then, we showcase several lifted inference approaches. When concluding, we touch upon the main challenge lying ahead, *asymmetry*.

2 Symmetries in Graphical Models

Let $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ be a set of n discrete-valued random variables and let x_i represent the possible realizations of random variable X_i. A graphical models compactly represents a joint distribution over $\mathbf{X}$ as a product of factors [46], i.e., $P(\mathbf{X} = \mathbf{x}) = Z^{-1} \prod_k f_k(\mathbf{x}_k)$ where each factor f_k is a non-negative function of a subset of the variables $\mathbf{x}_k$, and Z is a normalization constant. They can elegantly be represented using factor graphs. A factor graph, as shown in Fig. 2(Left), is a bipartite graph that expresses the factorization. It has a variable node (denoted as a circle) for each variable X_i, a factor node (denoted as a square) for each f_k, with an edge connecting variable node i to factor node k if and only if X_i is an argument of f_k.

As an example, consider the joint distribution $P(A, B, C)$ that has the factor graph given in Fig. 2(Right). For instance, we might be interested in distributing data to a network. Imagine that Anna, Bob and Charles participate in a peer-to-peer network where a file is divided into parts. Not all of them have all parts, and the nodes exchange these parts until they can re-assemble back to the complete file. Intuitively, there are three regions, namely $\{A, B\}$, $\{B, C\}$, and $\{B\}$, where region $\{B\}$ separates $\{A, B\}$ and $\{B, C\}$. That is, when conditioning on B, A and C are independent. Because of this, one can break down the full joint distribution as follows:

$$P(A,B,C) = P(C|A,B)P(A|B)P(B) \quad \textbf{(chain rule)}$$
$$= P(C|B)P(A,B) \quad \textbf{(}C\textbf{ independent of }A\textbf{ given }B\textbf{)}$$
$$= P(C|B)\frac{P(B)}{P(B)}P(A,B) \quad \textbf{(multiplication by }1\textbf{)}$$
$$= P(A,B)P(B,C)P(B)^{-1} \quad \textbf{(chain rule)}$$

Following Yedidia *et al.* [70], we visualize these regions of nodes in terms of a region graph as shown in Fig. 2(Middle). The number c_R associated with each region R is the so-called *counting number* and coincides in our example with the exponent of the corresponding term in the last equation. Thus, $P(\mathbf{X} = \mathbf{x}) = \prod_R P_R(\mathbf{x}_R)^{c_R}$ where P_R is the joint distribution of a region R, in our case $P(A, B)$, $P(B, C)$, and $P(B)$.

Imagine now that the model obeys to some additional symmetry, say, $P(a, b) = P(b, c)$ holds for a particular joint state a, b, c. For instance, Bob requires a file part that both Anna and Bob can provide. Then, the joint distribution simplifies to

$$P(a, B, c) = P(a, B)^2 P(B)^{-1} \quad \textbf{(symmetry)}$$

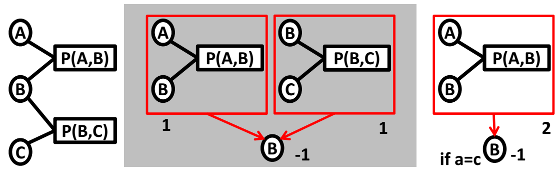

Figure 2. An example for symmetries in a graphical model. (Left) A factor graph encoding a joint distribtion $P(A, B, C)$. (Middle) The corresponding region graph where each region is encoded as a red box. (Right) Lifted region graph assuming $P(a, B) \equiv P(B, c)$. (Best viewed in color)

where we have used the upper-case B do stress that this holds for any state of B. This reflects our intuition. Bob can get the missing file part from both Anna and Charles; they are indistinguishable for Bob. Thus, the original model is "over-sized" in the sense that there are several joint states where only two instead of three terms are required. In these cases, we can compute the joint probability more efficiently: we compute $P(a, B)$ once and raise it to the power of 2, since it coincides with $P(B, c)$.

This can also be reflected in the region graph. As shown in Fig. 2(Right), since $P(a, B) = P(B, c)$, we group together both regions and associated the sum of the original counting numbers as counting number with the resulting superregion[3]. Since A and C take on the same state, the region graph and its lifted counterpart encode the same joint probability but using different regions and counting numbers, namely $\mathbf{c} = (1, 1, -1)$ **(ground)** and $\mathbf{c} = (2, -1)$ **(lifted)** where $\mathbf{c}$ is the corresponding vector of counting numbers. The different counting numbers directly translates to less computations in the lifted case.

It is important to note, however, that this only holds for the symmetric case. If A and C take on different states, we cannot group $P(a, B)$ and $P(B, c)$ together since $P(a, B) \neq P(B, c)$ in general. For instance, if Anna and Charles provide different parts, and Bob requires only one of them. In this asymmetric case, we still have to evaluate all three terms:

$$P(a, B, c) = P(a, B)P(B, c)P(B)^{-1}. \quad \textbf{(asymmetry)}$$

This simple insight is important. It illustrates that lifting is not always beneficial. There are simply asymmetric situations. In fact, Erdös and Rényi showed that almost all large graphs are asymmetric [18], but it is readily observed that many graphs representing structures of real interest contain symmetry. Generally, it is difficult to preserve the full joint distribution by lifting. Consequently, lifting approaches do not (yet) lift the model but rather specific inference tasks and algorithms.

In the following, we review[4] some lifted inference approaches. We distinguish between two classes: *bottom-up* approaches start from a given propositional model whereas *top-down* approaches start from a specification of a probabilistic model in first-order logical format.

[3] Note that the lifted region graph does not qualify as region graph anymore since it validates the *region graph condition*, see [70] for more details. The important point here is that the probability of joint states that conform to the symmetry is still correct since we start from a valid region graph.

[4] We would like to note that we do not touch upon methods for preprocessing [57, 38, 63] and lazy inference [49, 52] that can also reduce the running time of inference drastically, upon theoretical results on lifted inference [34, 29, 65], nor upon the use of lifted inference for relational probabilistic conditional logic [62] and for solving relational MDPs, see e.g. [53, 32]. We also do not touch upon graphical models with symmetries outside the SRL/StarAI context such as [5, 36, 19, 20].

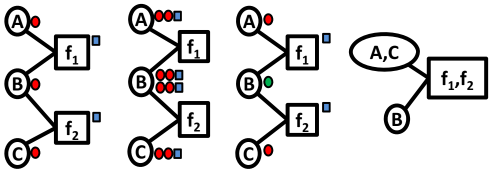

Figure 3. From left to right, the steps of lifting the factor graph in Fig. 1(Left). The colored small circles and squares denote the groups and signatures produced running color passing.

Note, however, that in many cases a bottom-up approach can be turned into a top-down approach and vice versa. A good example is lifted (loopy) belief propagation (that we will discuss next) for which both top-down [58] and bottom-up [30] variants exist.

3 Bottom-up Lifting: Lifted Inference for Propositional Models

An important inference task is to compute the conditional probability of each variable given the values of some others, the evidence, by summing out the remaining variables. The belief propagation (BP) algorithm [46] is an efficient way to solve this problem that is exact when the factor graph is a tree, but only approximate when the factor graph has cycles. One should note that the problem of computing marginal probability functions is in general hard (#P-complete).

The BP algorithm[5] sends messages between variable nodes and their neighboring factors (and vice versa) until convergence. Specifically, the message from a variable X to a factor f is

$$\mu_{X\to f}(x) = \prod\nolimits_{h\in \mathrm{nb}(X)\setminus\{f\}} \mu_{h\to X}(x)$$

where $\mathrm{nb}(X)$ is the set of factors X appears in. The message from a factor to a variable is

$$\mu_{f\to X}(x) = \sum\nolimits_{\neg\{X\}} \left(f(\mathbf{x}) \prod\nolimits_{Y\in \mathrm{nb}(f)\setminus\{X\}} \mu_{Y\to f}(y) \right)$$

where $\mathrm{nb}(f)$ are the arguments of f, and the sum is over all of these except X, denoted as $\neg\{X\}$. Initially, the messages are set to 1. After convergence or predefined number of iterations, the unnormalized belief of each variable X_i can be computed from the equation $b_i(x_i) = \prod_{f\in \mathrm{nb}(X_i)} \mu_{f\to X_i}(x_i)$.

Although already highly efficient, BP does not make use of symmetries. Reconsider our example shown Fig. 2. To exploit the symmetries present in the graph structure, *lifted* BP variants [58, 30], (that build upon [28]) essentially perform two steps: Given a factor graph, they first compute a lifted factor graph and then run a modified BP on it. In the first step, we simulate BP keeping track of which nodes and factors send the same messages, and group nodes and factors together correspondingly. In the first step, initially, all variable nodes and all identical factors fall into corresponding groups as indicated by the colors in Fig. 3. Now, each variable node sends its color to its neighboring factor nodes. A factor node collects the incoming colors, puts its own color at the end, cf. Fig. 3, and sends this color signature back to the neighboring variables nodes. The variable nodes stack the incoming signatures together and, hence, form unique signatures of their one-step message history. We group together variable nodes with the same message history and assign new colors to each group. The factors are grouped in a similar way. This color-passing process is iterated until no new colors are created anymore. At convergence, all variables nodes with the same color form a *supernode* and all factors with the same color a *superfactor*. In our case, variable nodes A, C and factor nodes f_1, f_2 are grouped together as shown in Fig. 3.

[5] We assume that any evidence is incorporated into the model by setting $f(\mathbf{x}) = 0$ for states $\mathbf{x}$ that are incompatible with it.

Since supernodes and -factors are sets of nodes and factors that send and receive the same messages at each step of carrying out BP, we can now simulate BP on the lifted factor graph. An edge in the lifted graph essentially represents multiple edges in the original factor graph. Let $c(\mathfrak{f}, \mathfrak{X}_i)$ be the number of identical messages that would be sent from the factors in the superfactor $\mathfrak{f}$ to each node in the supernode $\mathfrak{X}_i$ if BP was carried out on original factor graph. The message from a supernode $\mathfrak{X}$ to a superfactor $\mathfrak{f}$ is

$$\mu_{\mathfrak{X}\to\mathfrak{f}}(x) = \mu_{\mathfrak{f}\to\mathfrak{X}}(x)^{c(\mathfrak{f},\mathfrak{X})-1} \cdot \prod\nolimits_{\mathfrak{h}\in \mathrm{nb}(\mathfrak{X})\setminus\{\mathfrak{f}\}} \mu_{\mathfrak{h}\to\mathfrak{X}}(x)^{c(\mathfrak{h},\mathfrak{X})}$$

where $\mathrm{nb}(\mathfrak{X})$ now denotes the neighbor relation of supernode $\mathfrak{X}$ in the lifted graph. The $c(\mathfrak{f}, \mathfrak{X}) - 1$ exponent reflects the fact that a supernode's message to a superfactor excludes the corresponding factor's message to the variable if BP was carried out on the original factor graph. Likewise, the unnormalized belief of any random variable X in $\mathfrak{X}_i$ can be computed as follows $b_i(x_i) = \prod_{\mathfrak{f}\in \mathrm{nb}(\mathfrak{X}_i)} \mu_{\mathfrak{f}\to\mathfrak{X}_i}(x_i)^{c(\mathfrak{f},\mathfrak{X})}$.

However, as the original BP algorithm, lifted BP also does not prescribe a way to solve more complex inference tasks such as computing joint marginals for k-tuples of distant random variables or satisfying assignments of CNFs. A popular solution in these cases is the idea of turning the complex inference task into a sequence of simpler ones by selecting and clamping variables one at a time and running lifted message passing again after each selection. This naive solution, however, recomputes the lifted network in each step from scratch, therefore often canceling the benefits of lifted inference. Online lifting approaches avoid this by reusing already known liftings when computing the lifting of the next inference task [1, 42, 25] and can also be used to realize lifted sampling.

Lifted BP approaches are also appealing because they are simple, efficient, and parallelizable. Moreover, they have paved the way for lifted solutions of many important AI tasks. For instance, one can lift variants of BP for solving satisfiability problems such as survey propagation [26] or when the underlying distributions are Gaussian [3]. In turn, one can realize lifted variants of Kalman filters, PageRank, Label Propagation, and Clustering-on-demand [3, 43]. Even linear programming solvers can be lifted. Intuitively, given a linear program, we employ a lifted variant of Gaussian BP to solve the systems of linear equations arising when running an interior-point method to solve the linear program. However, this naive solution cannot make use of standard solvers for linear equations and is doomed to construct lifted networks in each iteration of the interior-point method again, an operation that can itself be quite costly. Mladenov *et al.* [41] showed that we can read off an equivalent linear program from the lifted Gaussian BP computations that can be solved using any off-the-shelf linear program solver. More importantly, this connects lifted inference and linear program relaxations for the MAP inference problem, see e.g. [22].

In a distinct yet related work, Sen *et al.* [55] proposed the idea of bisimulated variable elimination (VE). In a nutshell, VE [73] works as follows. To compute a single node marginal, we iterate the following steps: we select a random variable X, multiply all factors

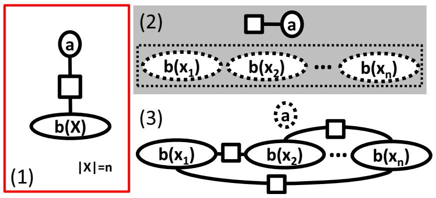

Figure 4. An example for lifted first-order variable elimination. (1) A parfactor consists of two atoms a and b(X). (2) When summing out a, all ground instances of b(X) can be grouped together. (3) The case when eliminating b(X). All ground instances of b(X) are now connected. (Best viewed in color)

together in which X appears, sum out X from the product, and store the result. To lift VE, Sen *et al.* used the resulting computation trees (i.e., the tree-structured "unrolling" of the graphical model computations rooted at the nodes) when running VE to group together individuals indistinguishable. To identify these indistinguishable individuals, they employ bisimulation. In general, given a graph and some desired property, bisimulation algorithms partition the set of vertices into disjoint sets such that every pair of vertices in each set satisfies the property. The property that Sen *et al.* identify is whether two factors have the same input and output values when running VE. This idea was later extended by the same group to approximate inference [56]. Essentially, they trade off inference accuracy for computational efficiency, i.e., lifting by e.g. grouping nodes and factors together that are within a user-specified ϵ-distance of each other. Similar ideas have been explored within lifted BP [31, 59].

4 Top-Down Lifting: Lifted Inference for Relational Models

Although bottom-up lifting can be applied to models for which a relational representation does not exist or is not the most intuitive way of encoding, the more natural case for lifted inference are relational models. If a relational model is given, we can seek inspiration from logical inference, where lifted inference such as resolution is commonly performed.

That is, we also start with the model and refine it (also called shattering) until indistinguishable nodes and factors form groups. Essentially, we start with a set of parameterized factors or *parfactors* [47]. A parfactor is a triple $\langle C, V, t\rangle$ where C is a set of inequality constraints on logical variables, V is a set of parameterized random variables and t is a factor on V. Note that t will be the factor that is used for all assignments of individuals to logical variables (placeholders) in V. If the factor represents a clause, the table specifies just a single number, and then V is written as a first-order clause as for example done in MLNs. As an example, also illustrated in Fig. 4(1), consider a parfactor on $\langle\{\}, \{\mathtt{a}, \mathtt{b}(\mathtt{X})\}, t\rangle$ where the population of X, i.e., the possible assignments of individuals to the logical variables X has size n. Intuitively, in our data distribution task, this corresponds to the case that all b(X) can provide file parts to a. Our first task is to compute $P(\mathtt{a})$, i.e., we want to know from whom a is going to request a missing file part. To do so, we have to sum out all instances of b(X). When summing out all instances of b(X), see also Fig. 4(2), we can note that all of the factors in the grounding have the same value (since all instance of b(X) provide the same file parts) and so can be taken to the power of n, which can be done in time logarithmic in n, whereas the grounding is linear in n. This operation, invented by David Poole [47], has been called *inversion elimination* by [14]. However, if we were to sum out a instead, see Fig. 4(3), in the resulting grounding all instances of b(X) are connected, and so there would be a factor that is of size exponential in n. de Salvo Braz *et al.* [14] showed how, rather than representing the resulting factor, we only need to count the number of instances of b(X), which have a certain value, and so the subsequent elimination of b(X) can be done in time polynomial in n. This is called *counting elimination* and is linear in n if b(X) is binary, and if b(X) has k values, the time is $O(n^{k-1})$. Both elimination operations are restricted in different ways. Inversion elimination operates in time independent from domain size, but can eliminate an atom only if it contains all logical variables in its parfactor, and its grounding is disjoint from any other atoms in that parfactor. Counting elimination can deal with atoms whose grounding is the same as some other atom in the parfactor, but logical variables in one atom cannot be constrained by those of others. In any case, we can now lift variable elimination (VE) by repeating the following steps: (1) Pick a parameterized variable X. (2) If applicable, apply inversion respectively counting elimination to sum out the whole group of random variables represented by X. Store the resulting new parfactors. (3) If neither of them is applicable, shatter two parfactors X is involved. This splits each parfactor into a part that is shared with the other parfactor, and a part that is disjoint, hopefully making one of them applicable in the next round.

For propositional models, lifted VE essentially coincides with VE. However, by avoiding many redundant computations, lifted VE can achieve an exponential speedup compared to VE for relational models, moving essentially from $\mathcal{O}(2^n)$ to $\mathcal{O}(n)$; ultimately being independent of the domain size. Of course, in a similar way, one can lift MAP and MPE inference approaches based on VE [15]. Even more efficiency can be gained if we not only employ sharing of potentials across interchangeable random variables but also to exploit interchangeability within individual potentials. To do so, Milch *et al.* [40] proposed counting formulas as a representation of the intermediate lifted formulas. They indicate how many of the random variables in a set have each possible value. Because counting formulas capture additional symmetries among large numbers of variables compactly, this can result in asymptotic speed improvements compared to de Salvo Braz *et al.*'s approach. Kisyński and Poole [35] have shown how to perform lifted inference within directed first-order models that require an aggregation operator when a parent random variable is parameterized by logical variables that are not present in a child random variable. Recently, Choi *et al.* [9] have shown how to perform lifted VE in the presence of aggregate factors such as SUM, AVERAGE, and AND in probabilistic relational models. Taghipour *et al.* [61] have lifted the restriction of having inequality constraints only towards arbitrary constraints and report even more speed ups. Choi *et al.* [8] addressed lifted VE when the underlying distributions are Gaussian. Their approach assumes that the model consists of Gaussian potentials. Their algorithm marginalizes variables by integrating out random variables using inversion elimination operation. If the elimination is not possible, they consider elimination of pairwise potentials and the marginals that are not in pairwise form are converted to pairwise form and then eliminated. Recently, the same group has shown how to realize a lifted Kalman filter based on lifted VE [10]. Recently, Van den Broeck *et al.* [66] built a bridge between lifted VE and lifted BP by lifting the "relax, compensate and then recover" [7].

An alternative to variable elimination is to use search-based methods based on recursive conditioning. That is, we decompose by conditioning on parameterized variables a lifted network into smaller networks that can be solved independently. Each of these subnetworks is then solved recursively using the same method, until we reach a simple enough network that can be solved [11]. Recently, several top-down lifted search-based methods have been proposed [23, 24, 68, 48]. Gogate and Domingos [24] reduced the problem of lifted probabilistic inference to weighted model counting in a lifted graph. Van den Broeck *et al.* [68] employ circuits in first-order deterministic decomposable negation normal form to do the same, also for higher order marginals [67]. Both these approaches were developed in parallel and have promising potential to lifted inference.

Finally, there are also sampling methods that employ ideas of lifting. Milch and Russell developed an MCMC approach where states are only partial descriptions of possible worlds [39]. Zettlemoyer *et al.* [72] extended particle filters to a logical setting. Gogate and Domingos introduced a lifted importance sampling [24]. Recently, Niepert proposed permutation groups and group theoretical algorithms to represent and manipulate symmetries in probabilistic models, which can be used for MCMC [44].

5 An "Asymmetric" Conclusion

We have seen several lifted inference approaches. However, already in 1848, Louis Pasteur recognized "Life as manifested to us is a function of the asymmetry of the universe". This remark characterizes somehow one of the main challenges for lifted probabilistic inference: Not only are almost all large graphs asymmetric [18], but *even if there are symmetries within a probabilistic model, they easily break when it comes to inference since variables become correlated by virtue of depending asymmetrically on evidence*. This, however, does not mean that lifted inference is hopeless. Indeed, in many cases lifting will produce a new model that is not far from propositionalized, therefore canceling the benefits of lifted inference. However, in many asymmetric cases we can do considerably better. de Salvo Braz *et al.* [16] presented anytime lifted BP. It performs shattering during BP inference, on an as-needed basis, starting on the most relevant parts of a model first. The trade-off is having an (exact) bound (an interval) on the query's belief rather than an exact belief. Or, we may use existing approximate lifting such as [56, 31, 59]. We may also use sequences of increasingly fine approximations to control the trade-off between lifting and accuracy [33]. Another appealing idea, in particular when learning the parameters of relational models, is to break the global model in to local ones and then to train and recombine the local models. This breaks long-range dependencies and allows to exploit lifting within and across the local training tasks [2]. Recently, Bui *et al.* [6] have shown that for MAP inference we can exploit the symmetries of the model before evidence is obtained.

To conclude, one of the key challenges in building intelligent agents is closing the gap between logical and statistical AI, so that we can have rich representations including objects, relations and uncertainty, that we can effectively learn and carry out inference with. Real agents need to deal with their uncertainty and reason about individuals and relations. They need to learn how the world works before they have encountered all the individuals they need to reason about. Over the last 25 years there has been a considerable body of research into combinations of predicate logic and probability forming what has become known as statistical relational artificial intelligence (StarAI). However, if we accept the premises of StarAI, then we need to get serious about lifted probabilistic inference and learning. While there have been considerable advances already, there are more than enough problems, in particular asymmetric ones, to go around to really establish what has come to be called statistical relational AI.

ACKNOWLEDGEMENTS

The author would like to thank Sriraam Natarajan and David Poole for a recent joint effort in writing an introduction to statistical relational AI, parts of which grew into the present paper. The author is also grateful to all his collaborators and the SRL/StarAI community for the interesting discussions and inspirations. Thanks! This work was supported by the Fraunhofer ATTRACT fellowship STREAM and by the EC, FP7-248258-First-MM.

REFERENCES

[1] B. Ahmadi, K. Kersting, and F. Hadiji, 'Lifted belief propagation: Pairwise marginals and beyond', in *Proc. of the 5th European WS on Probabilistic Graphical Models (PGM–10)*, Helsinki, Finland, (2010).

[2] B. Ahmadi, K. Kersting, and S. Natarajan, 'Lifted online training of relational models with stochastic gradient methods', in *Proc. of the European Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, (2012).

[3] B. Ahmadi, K. Kersting, and S. Sanner, 'Multi-Evidence Lifted Message Passing, with Application to PageRank and the Kalman Filter', in *Proc. of the 22nd Int. Joint Conf. on AI (IJCAI–11)*, (2011).

[4] R. Bödi, K. Herr, and M. Joswig, 'Algorithms for highly symmetric linear and integer programs', *Mathematical Programming, Series A*, (Online First, Jan. 2011).

[5] S.P. Boyd, P. Diaconis, P.A. Parrilo, and L. Xiao, 'Fastest mixing markov chain on graphs with symmetries', *SIAM Journal on Optimization*, **20**(2), 792–819, (2009).

[6] H.H. Bui, T. Huynh, and R. de Salvo Braz, 'Exact lifted inference with distinct soft evidence on every object', in *Proc. of the 26th AAAI Conf. on Artificial Intelligence (AAAI 2012)*, (2012).

[7] A. Choi and A. Darwiche, 'Relax, compensate and then recover', in *New Frontiers in Artificial Intelligence*, eds., Takashi Onada, Daisuke Bekki, and Eric McCready, volume 6797 of *Lecture Notes in Computer Science*, 167–180, Springer Berlin / Heidelberg, (2011).

[8] J. Choi and E. Amir, 'Lifted inference for relational continuous models', in *Proc. of the 26th Conf. on Uncertainty in Artificial Intelligence (UAI-10)*, (2010).

[9] J. Choi, R. de Salvo Braz, and H.H. Bui, 'Efficient methods for lifted inference with aggregate factors', in *Proc. of the 25th AAAI Conf. on Artificial Intelligence (AAAI 2011)*, (2011).

[10] J. Choi, A. Guzman-Rivera, and E. Amir, 'Lifted relational kalman filtering', in *Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence(IJCAI)*, pp. 2092–2099, (2011).

[11] Adnan Darwiche, 'Recursive conditioning', *Artificial Intelligence*, **126**(1-2), 5–41, (2001).

[12] L. De Raedt, *Logical and Relational Learning*, Springer, 2008.

[13] *Probabilistic Inductive Logic Programming*, eds., L. De Raedt, P. Frasconi, K. Kersting, and S.H. Muggleton, volume 4911 of *Lecture Notes in Computer Science*, Springer, 2008.

[14] R. de Salvo Braz, E. Amir, and D. Roth, 'Lifted First Order Probabilistic Inference', in *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05)*, pp. 1319–1325, (2005).

[15] R. de Salvo Braz, E. Amir, and D. Roth, 'Mpe and partial inversion in lifted probabilistic variable elimination', in *Proc. of the 21sth National Conf. on Artificial Intelligence (AAAI)*, (2006).

[16] R. de Salvo Braz, S. Natarajan, H. Bui, , J. Shavlik, and S. Russell, 'Anytime lifted belief propagation', in *Working Notes of the Int. Workshop on Statistical Relational Learning (SRL-09)*, (2009).

[17] T. Dean and R. Givan, 'Model minimization in markov decision processes', in *Proc. of the Fourteenth National Conf. on Artificial Intelligence (AAAI-97)*, pp. 106–111, (1997).

[18] P. Erdös and A. Rényi, 'Asymmetric graphs', *Acta Math. Acad. Sci. Hungar.*, **14**, 295–315, (1963).

[19] H. Gehrmann, 'Lattices of graphical gaussian models with symmetries', *Symmetry*, **3**(3), 653–679, (2011).

[20] H. Gehrmann and S.L. Lauritzen, 'Estimation of means in graphical gaussian models with symmetries', *CoRR*, **abs/1101.3709**, (2011).

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

[22] A. Globerson and T. Jaakkola, 'Fixing max-product: Convergent message passing algorithms for map LP-relaxations', in *Proc. of the 21st Annual Conf. on Neural Inf. Processing Systems (NIPS)*, (2007).

[23] V. Gogate and P. Domingos, 'Exploiting logical structure in lifted probabilistic inference', in *Working Note of the AAAI-10 Workshop on Statistical Relational Artificial Intelligence*, (2010).
[24] V. Gogate and P. Domingos, 'Probabilistic theorem proving', in *Proc. ot the 27th Conf. on Uncertainty in Artificial Intelligence (UAI)*, (2011).
[25] F. Hadiji, B. Ahmadi, and K. Kersting, 'Efficient sequential clamping for lifted message passing', in *Proc. of the 34th Annual German Conf. on Artificial Intelligence (K–11)*, pp. 122–133, (2011).
[26] F. Hadiji, K. Kersting, and B. Ahmadi, 'Lifted message passing for satisfiability', in *Working Notes of the AAAI10 Workshop on Statistical Relational AI (StarAI)*. AAAI Press, (2010).
[27] K. Herr and R. Bödi, 'Symmetries in linear and integer programs', *CoRR*, **abs/0908.3329**, (2010).
[28] A. Jaimovich, O. Meshi, and N. Friedman, 'Template-based Inference in Symmetric Relational Markov Random Fields', in *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI-07)*, pp. 191–199, (2007).
[29] A.K. Jha, V. Gogate, A. Meliou, and D. Suciu, 'Lifted inference seen from the other side : The tractable features', in *Proc. of the 24th Annual Conf. on Neural Information Processing Systems (NIPS-2010)*, pp. 973–981, (2010).
[30] K. Kersting, B. Ahmadi, and S. Natarajan, 'Counting Belief Propagation', in *Proc. of the 25th Conf. on Uncertainty in Artificial Intelligence (UAI–09)*, (2009).
[31] K. Kersting, Y. El Massaoudi, B. Ahmadi, and F. Hadiji, 'Informed lifting for message–passing', in *Proc. of the 24th AAAI Conf. on Artificial Intelligence (AAAI–10)*, (2010).
[32] R. Khardon, 'Stochastic planning and lifted inference', in *Working Notes of the AAAI-10 Workshop on Statistical Relational Artificial Intelligence*, (2010).
[33] C. Kiddon and P. Domingos, 'Coarse-to-fine inference and learning for first-order probabilistic models', in *Proc. of the 25th AAAI Conf. on Artificial Intelligence (AAAI 2011)*, (2011).
[34] J. Kisyński and D. Poole, 'Constraint processing in lifted probabilistic inference', in *Proc. of the 25th Conf. on Uncertainty in Artificial Intelligence (UAI-2009)*, pp. 293–302, (2009).
[35] J. Kisyński and D. Poole, 'Lifted aggregation in directed first-order probabilistic models', in *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pp. 1922–1929, (2009).
[36] P. Liang, M.I. Jordan, and D. Klein, 'Type-based mcmc', in *Proc. of the Conf. of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL-2010)*, pp. 573–581, (2010).
[37] F. Margot, 'Symmetry in integer linear programming', in *50 Years of Integer Programming 1958–2008: From the Early Years to the State-of-the-Art*, eds., M. Jünger, T.M. Liebling, D. Naddef, G.L. Nemhauser, W.R. Pulleyblank, G. Reinelt, G. Rinaldi, and L.A. Wolsey, 1–40, Springer, (2010).
[38] L. Mihalkova and M. Richardson, 'Speeding up inference in statistical relational learning by clustering similar query literals', in *Proc. of the 19th Int. Conf. on Inductive Logic Programming (ILP-09)*, (2009).
[39] B. Milch and S.J. Russell, 'General-purpose mcmc inference over relational structures', in *Proc. of the 22nd Conf. in Uncertainty in Artificial Intelligence (UAI-2006)*, (2006).
[40] B. Milch, L. Zettlemoyer, K. Kersting, M. Haimes, and L. Pack Kaelbling, 'Lifted Probabilistic Inference with Counting Formulas', in *Proc. of the 23rd AAAI Conf. on Artificial Intelligence (AAAI-08)*, (2008).
[41] M. Mladenov, B. Ahmadi, and K. Kersting, 'Lifted linear programming', in *15th Int. Conf. on Artificial Intelligence and Statistics (AISTATS 2012)*, pp. 788–797, (2012). Volume 22 of JMLR: W&CP 22.
[42] A. Nath and P. Domingos, 'Efficient lifting for online probabilistic inference', in *Proc. of the 24th AAAI Conf. on Artificial Intelligence (AAAI-10)*, (2010).
[43] M. Neumann, K. Kersting, and B. Ahmadi, 'Markov logic sets: Towards lifted information retrieval using pagerank and label propagation', in *Proc. of the 25th AAAI Conf. on Artificial Intelligence (AAAI–11)*, (2011).
[44] M. Niepert, 'Markov chains on orbits of permutation groups', in *Proc. of the 28th Conf. on Uncertainty in Artificial Intelligence (UAI)*, (2012).
[45] N. Nilsson, 'Probabilistic Logic', *Artificial Intelligence*, **28**, 71–87, (1986).
[46] J. Pearl, *Reasoning in Intelligent Systems: Networks of Plausible Inference*, MK, 1991.
[47] D. Poole, 'First-Order Probabilistic Inference', in *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI-05)*, pp. 985–991, (2003).
[48] D. Poole, F. Bacchus, and J. Kisyński, 'Towards completely lifted search-based probabilistic inference', *CoRR*, **abs/1107.4035**, (2011).
[49] H. Poon, P. Domingos, and M. Sumner, 'A general method for reducing the complexity of relational inference and its application to mcmc', in *Proc. of the 23rd National Conf. on Artificial Intelligence (AAAI-2008)*, pp. 1075–1080, (2008).
[50] B. Ravindran and A.G. Barto, 'Symmetries and model minimization in markov decision processes', Technical Report 01-43, University of Massachusetts, Amherst, MA, USA, (2001).
[51] M. Richardson and P. Domingos, 'Markov Logic Networks', *Machine Learning*, **62**, 107–136, (2006).
[52] S. Riedel, 'Improving the accuracy and efficiency of map inference for markov logic', in *Proc. of the 24th Conf. in Uncertainty in Artificial Intelligence (UAI-2008)*, pp. 468–475, (2008).
[53] S. Sanner and C. Boutilier, 'Practical solution techniques for first-order MDPs', *Artif. Intell.*, **173**, 748–488, (2009).
[54] M. Sellmann and P. Van Hentenryck, 'Structural symmetry breaking', in *Proc. of 19th International Joint Conf. on Artificial Intelligence (IJCAI-05)*, (2005).
[55] P. Sen, A. Deshpande, and L. Getoor, 'Exploiting shared correlations in probabilistic databases', *Proc. VLDB Endow.*, **1**, 809–820, (2008).
[56] P. Sen, A. Deshpande, and L. Getoor, 'Bisimulation-based approximate lifted inference', in *Uncertainty in Artificial Intelligence*, (2009).
[57] J. Shavlik and S. Natarajan, 'Speeding Up Inference in Markov Logic Networks by Preprocessing to Reduce the Size of the Resulting Grounded Network', in *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI-05)*, pp. 1951–1956, (2009).
[58] P. Singla and P. Domingos, 'Lifted First-Order Belief Propagation', in *Proc. of the 23rd AAAI Conf. on Artificial Intelligence (AAAI-08)*, pp. 1094–1099, Chicago, IL, USA, (July 13-17 2008).
[59] P. Singla, A. Nath, and P. Domingos, 'Approximate lifted belief propagation', in *Working Notes of the AAAI-10 Workshop on Statistical Relational Artificial Intelligence*, (2010).
[60] I. Stewart, *Why Beauty Is Truth: A History of Symmetry*, Basic Books, 2007.
[61] N. Taghipour, D. Fierens, J. Davis, and H. Blockeel, 'Lifted variable elimination with arbitrary constraints', in *15th Int. Conf. on Artificial Intelligence and Statistics (AISTATS 2012)*, pp. 1194–1202, (2012). Volume 22 of JMLR: W&CP 22.
[62] M. Thimm, 'On prototypical indifference and lifted inference in relational probabilistic conditional logic', in *Proc. of the Third Workshop on Dynamics of Knowledge and Belief (DKB-2011) at KI-2011*, eds., C. Beierle and G. Kern-Isberner, pp. 1–15, (2011).
[63] I. Thon, N. Landwehr, and L. De Raedt, 'Stochastic relational processes: Efficient inference and applications', *Machine Learning*, **82**(2), 239–272, (2011).
[64] G. Van den Broeck, 'On the completeness of first-order knowledge compilation for lifted probabilistic inference', in *Proceedgins of the 25th Annual Conf. on Neural Information Processing Systems (NIPS-2011)*, pp. 1386–1394, (2011).
[65] G. Van den Broeck, A. Choi, and A. Darwiche, 'Lifted relax, compensate and then recover: From approximate to exact lifted probabilistic inference', in *Proc. of the 28th Conf. on Uncertainty in Artificial Intelligence (UAI)*, (2012).
[66] G. Van den Broeck and J. Davis, 'Conditioning in first-order knowledge compilation and lifted probabilistic inference', in *Proc. of the 26th AAAI Conf. on Artificial Intelligence (AAAI-2012)*, (2012).
[67] G. Van den Broeck, N. Taghipour, W. Meert, J. Davis, and L. De Raedt, 'Lifted probabilistic inference by first-order knowledge compilation', in *Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pp. 2178–2185, (2011).
[68] T. Walsh, 'Symmetry breaking constraints: Recent results', in *Proc. of the 26th AAAI Conf. on Artificial Intelligence (AAAI-2012)*, (2012).
[69] H. Weyl, *Symmetry*, Princeton University Press, 1952.
[70] J.S. Yedidia, W.T. Freeman, and Y. Weiss, 'Constructing free-energy approximations and generalized belief propagation algorithms', *IEEE Transactions on Information Theory*, **51**(7), 2282–2312, (2005).
[71] D.W. Zaidel and M. Hessamian, 'Asymmetry and symmetry in the beauty of human faces', *Symmetry*, **2**, 136–149, (2010).
[72] Luke S. Zettlemoyer, Hanna M. Pasula, and Leslie Pack Kaelbling, 'Logical particle filtering', in *Proc. of the Dagstuhl Seminar on Probabilistic, Logical, and Relational Learning*, (2007).
[73] N.L. Zhang and D. Poole, 'A simple approach to Bayesian network computations', in *Proc. 10th Canadian Conf. on AI*, pp. 171–178, (1994).