Towards High-Level Probabilistic Reasoning with Lifted Inference

Guy Van den Broeck
Department of Computer Science
KU Leuven, Belgium
guy.vandenbroeck@cs.kuleuven.be

Abstract

High-level representations of uncertainty, such as probabilistic logics and programs, have been around for decades. Lifted inference was initially motivated by the need to make reasoning algorithms high-level as well. While the lifted inference community focused on machine learning applications, the high-level reasoning goal has received less attention recently. We revisit the idea and look at the capabilities of the latest techniques in lifted inference. This lets us conclude that lifted inference is strictly more powerful than propositional inference on high-level reasoning tasks.

Introduction

When Poole (2003) originally conceived lifted inference, it was motivated by the following question. *"Suppose we observe that (i) Joe has purple hair, a purple car, and has big feet, and (ii) a person with purple hair, a purple car, and who is very tall was seen committing a crime. What is the probability that Joe is guilty?"* Despite its simplicity, this problem is prevalent in criminal trials: it is an example of the *prosecutor's fallacy* (Thompson and Schumann 1987). To represent the problem, several high-level languages have been proposed (Richardson and Domingos 2006; Milch et al. 2007). To solve it, one needs to know the probability of purple hair, a purple car, big feet, but most importantly, the size of the population that Joe is part of. Poole (2003) already proposed an algorithm that performs high-level reasoning for this problem, factoring in the populations size using exponentiation – an operation that is not supported by classical algorithms.

Since Poole's original question, lifted inference has made much progress, but the focus has shifted. Current research looks at machine learning applications (Ahmadi, Kersting, and Natarajan 2012; Van Haaren et al. 2014), large-scale probabilistic databases (Gribkoff, Suciu, and Van den Broeck 2014), exploiting symmetries for approximate inference (Niepert 2013; Mladenov, Globerson, and Kersting 2014; Bui, Huynh, and Sontag 2014), and even exploiting approximate symmetries (Van den Broeck and Darwiche 2013; Singla, Nath, and Domingos 2014; Venugopal and Gogate 2014; Van den Broeck and Niepert 2015). The overall goal of those works is to bring lifted inference to the mainstream, by showing it speeds up traditional learning and reasoning tasks. The type of question that Poole asks is rarely considered.

The move away from these high-level questions is best exemplified by the experimental evaluations found in recent lifted inference papers, which typically involve Markov logic networks (MLNs) (Richardson and Domingos 2006) for collective classification, entity resolution, or social network analysis. These are powerful models for machine learning, but they are not interpretable. It is not clear why a computer should have high-level reasoning capabilities on these tasks and be able solve them efficiently. As humans, we certainly do not have that capability.[1]

The fact that lifted inference works on large and complex MLNs is impressive and useful, but it does not exactly demonstrate Poole's original vision of a new type of high-level reasoning task. In this paper, we discuss tasks that do demonstrate this vision, including a task that humans can easily solve, but which we show cannot be solved efficiently with traditional techniques. Next, we review recent advances in lifted inference that enable more high-level reasoning: weighted first-order model counting and its inference rules. We demonstrate the power of these advances on an example task, showing that lifted inference is strictly more powerful than propositional reasoning techniques.[2]

High-Level Reasoning Tasks

This section describes two classes of tasks that pose high-level probabilistic reasoning challenges. We will see problems whose textual description is very simple and that can be solved using elementary statistical techniques. However, no classical automated reasoning algorithms can apply the same techniques for general-purpose inference.

Reasoning about Populations Suppose we are epidemiologists investigating a rare disease that is expected to present itself in one in every billion people. The probabil-

[1] They express correlation, not causation (Fierens et al. 2012).

[2] A similar separation in complexity was shown for lifted query evaluation in probabilistic databases by Beame et al. (2013).

ity that somebody in the world has the disease is

$$1-(1-1/1000000000)^{7000000000} \approx 0.999. \quad (1)$$

Here, we have used the fact that, for the level of detail we have knowledge about, all people are indistinguishable and independent. Therefore, we can exponentiate the probability that a single person is healthy to obtain the probability that all people are healthy. When encoding this distribution as a probabilistic graphical model, in the best case, inference ends up multiplying 7 billion numbers. In the worst case, naively representing the query variable's dependency on each person's health already requires a conditional probability table with $2^{7000000000}$ rows. In either case, despite the simplicity of the problem, graphical model algorithms do not exploit the symmetries that we exploit in Equation 1.

Now consider more complex queries. The probability that exactly five people are sick is $\binom{7\cdot 10^9}{5} \cdot \left(1-10^{-9}\right)^{7\cdot 10^9-5} \cdot \left(10^{-9}\right)^5 \approx 0.13$. More than five are sick with probability

$$1-\sum_{n=0}^{5}\binom{7\cdot 10^9}{n}\left(1-10^{-9}\right)^{7\cdot 10^9-n}\left(10^{-9}\right)^n \approx 0.7.$$

Here, we have used the indistinguishability of people in both the exponentiation, and in the binomial coefficients. It is not clear how to solve this problem with graphical models, but it would likely require enumerating all possible worlds.

The next example illustrates how to group indistinguishable objects into equivalence classes and reason about these classes as a whole. This technique is essential in lifted inference. Suppose we know that the disease is more rare in women, presenting only in 1 in every 2 billion women and 1 in every billion men. Then, the probability that more than five are sick is $1-\sum_{n=0}^{5}\sum_{f=0}^{n} P(n,f)$ where $P(n,f)$ is

$$\binom{3.6\cdot 10^9}{f}\left(1-0.5\cdot 10^{-9}\right)^{3.6\cdot 10^9-f}\left(0.5\cdot 10^{-9}\right)^f$$
$$\cdot\binom{3.4\cdot 10^9}{(n-f)}\left(1-10^{-9}\right)^{3.4\cdot 10^9-(n-f)}\left(10^{-9}\right)^{(n-f)}.$$

Here, we have split up the computation into two factors, for women on the first line and men on the second. The variable n represents the number of people with the disease and the variable f counts the number of women among them.

Reasoning about Playing Cards Consider a randomly shuffled deck of 52 playing cards. Suppose that we are dealt the top card, and we want to answer the following basic questions. What is the probability that we get hearts? When the dealer makes a mistake and reveals that the bottom card is black, how does our probability change? Basic statistics tells us that the probability increases from $1/4$ to $13/51$.

To represent the distribution over all shuffled decks with a probabilistic graphical model, a natural choice is to have 52 random variables, one for every card. Each variable can take any of 52 values, one for every position in the deck. When the queen of hearts takes the top position, none of the other random variables are allowed to take that position. Such constraints are enforced by adding a factor between every pair of

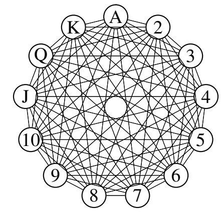

Figure 1: Graphical model for a shuffled deck of 13 cards

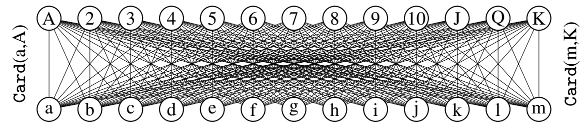

Figure 2: Bipartite graph from cards to positions $a,\ldots,m$

variables, setting the probability to zero that two cards are in the same position. Figure 1 depicts the graphical model for a small deck of 13 cards.

The graphical model for this problem is a completely connected graph whose treewidth grows linearly with the number of cards. This means that classical inference algorithms, such as junctions trees and variable elimination, will require time and space that is exponential in the number of playing cards (Darwiche 2009). Indeed, for a full deck of cards, these algorithms will build a joint probability table with 52^{52} rows. Of course, this makes our queries extremely intractable for classical algorithms. The underlying reason for this poor performance is that the distribution has no conditional or contextual independencies. Our belief about the top card is affected by any new observation on the remaining cards. The reason why humans can still answer the above queries efficiently is different: the distribution exhibits exchangeability (Niepert and Van den Broeck 2014).

The problem with this reasoning task is not one of representation. Many statistical relational languages, including Markov logic, can express the distribution concisely. It can even be written in classical first-order logic as

$$\begin{aligned} &\forall p, \exists c, \quad \texttt{Card}(p,c)\\ &\forall c, \exists p, \quad \texttt{Card}(p,c)\\ &\forall p, \forall c, \forall c', \neg\texttt{Card}(p,c) \vee \neg\texttt{Card}(p,c') \vee c=c', \end{aligned} \quad (2)$$

where $\texttt{Card}(p,c)$ denotes that position p contains card c.

Complexity of Propositonal Reasoning For Theory 2, we will now argue that no propositional inference algorithm can be efficient, unless P=NP. This is surprising, given how easily humans can reason about the card distribution. We do not require that the propositional inference algorithm comes with a treewidth-based lower complexity bound. We also do not require that the computation is isomorphic to a circuit language (as in Beame et al. (2013)). We instead assume that the propositional reasoner can efficiently deal with unary potentials (factors over a single variables). This assumption holds for all exact graphical model algorithms. Later, we discuss the performance of lifted inference.

Observe that the $\texttt{Card}$ relation represents a bipartite graph from cards to positions, where $\texttt{Card}(p, c)$ denotes an edge from c to p. Figure 2 shows such a graph. Theory 2 forces this bipartite graph to be a *perfect matching*, meaning that every card gets matched with a unique position. Hence, the number of possible words with non-zero probability equals the number of perfect matchings in a completely connected bipartite graph, which is $n!$. Note that Theory 2 can be extended with unary potentials to remove edges in Figure 2, by setting the probability of $\texttt{Card}$ atoms to zero. Then, the number of worlds with non-zero probability equals the number of perfect matchings in an arbitrary bipartite graph. Valiant (1979) proved that counting such matchings is #P-complete. Moreover, the number of worlds can be computed from the partition function of Theory 2, or from the marginal probabilities of a slightly modified representation. Therefore, such queries cannot be answered in polynomial time by a propositional algorithm, unless P=NP. A similar argument applies to other representations of the playing cards problems, including the graphical model of Figure 1.

Weighted First-Order Model Counting

This section highlights some recent developments in lifted inference that significantly increase the capabilities for high-level reasoning. These developments are centered around the weighted model counting (WMC) task. In model counting, or #SAT, one counts the number of assignments ω that satisfy a propositional sentence Δ, denoted $\omega \models \Delta$. In WMC, each assignment has a weight and the task is to compute the *sum of the weights* of all satisfying assignments.

Definition 1 (Weighted Model Count)**.** The WMC of

- a *sentence* Δ in propositional logic over literals $\mathcal{L}$, and
- a *weight function* $\mathrm{w} : \mathcal{L} \to \mathbb{R}$,

is defined as $\mathrm{WMC}(\Delta, \mathrm{w}) = \sum_{\omega \models \Delta} \prod_{l \in \omega} \mathrm{w}(l)$.

The success of WMC for inference in graphical models (Chavira and Darwiche 2008) has lead Van den Broeck et al. (2011) and Gogate and Domingos (2011) to propose *weighted first-order model counting* (WFOMC) as the core reasoning task underlying lifted inference algorithms. Instead of propositional logic, WFOMC works with theories Δ in finite-domain first-order logic. Another key assumption is that the function w in a WFOMC problem assigns identical weights to all positive (negative) literals of the same relation.

Several inference tasks in statistical relational models can be converted to WFOMC. This includes partition function and marginal computations in Markov logic, parfactors, probabilistic logic programs and relational Bayesian networks. We refer to Van den Broeck, Meert, and Darwiche (2013) for details. Its separation of logic and probability will enable complex logical transformations to be applied to Δ (also see Gribkoff, Suciu, and Van den Broeck (2014)).

Exponentiation We will now illustrate the principles behind WFOMC solvers. Algorithmic details can be found in Van den Broeck (2013). For the sake of simplicity, the examples are non-weighted model counting problems, corresponding to WFOMC problems where $\mathrm{w}(l) = 1$ for all literals $l \in \mathcal{L}$. Consider Δ to be

$$\texttt{Stress}(\mathsf{A}) \Rightarrow \texttt{Smokes}(\mathsf{A}). \tag{3}$$

Assuming a domain $\mathbf{D} = \{\mathsf{A}\}$, every assignment to $\texttt{Stress}(\mathsf{A})$ and $\texttt{Smokes}(\mathsf{A})$ satisfies Δ, except when $\texttt{Stress}(\mathsf{A})$ is true and $\texttt{Smokes}(\mathsf{A})$ is false. Therefore, the model count is 3. Now let Δ be

$$\forall x,\ \texttt{Stress}(x) \Rightarrow \texttt{Smokes}(x). \tag{4}$$

Without changing $\mathbf{D}$, the model count is still 3. When we expand $\mathbf{D}$ to n people, we get n independent copies of Formula 3. For each person x, $\texttt{Stress}(x)$ and $\texttt{Smokes}(x)$ can take 3 values, and the total model count is 3^n.

This example already demonstrates the benefits of first-order counting. A *propositional* model counter on Formula 4 would detect that all n clauses are independent, recompute for every clause that it has 3 models, and multiply these counts n times. Propositional model counters have no elementary operation for exponentiation. A *first-order* model counter reads from the first-order structure that it suffices to compute the model count of a single ground clause, and then knows to exponentiate. It never actually grounds the formula, and given the size of $\mathbf{D}$, it runs in logarithmic time. This gives an exponential speedup over propositional counting, which runs in linear time.

These first-order counting techniques can interplay with propositional ones. Take for example Δ to be

$$\forall y,\ \texttt{ParentOf}(y) \land \texttt{Female} \Rightarrow \texttt{MotherOf}(y). \tag{5}$$

This sentence is about a specific individual who may be female, depending on the proposition $\texttt{Female}$. We can separately count the models in either case. When $\texttt{Female}$ is false, Δ is satisfied, and the $\texttt{ParentOf}$ and $\texttt{MotherOf}$ atoms can take any value. This gives 4^n models. When $\texttt{Female}$ is true, Δ is structurally identical to Formula 4, and has 3^n models. The total model count is then $3^n + 4^n$.

These concepts can be applied recursively to count more complicated formulas. Take for example

$$\forall x, \forall y,\ \texttt{ParentOf}(x, y) \land \texttt{Female}(x) \Rightarrow \texttt{MotherOf}(x, y).$$

There is now a partition of the ground clauses into n independent sets of n clauses. The sets correspond to values of x, and the individual clauses to values of y. The formula for each specific x, that is, each set of clauses, is structurally identical to Formula 5 and has count of $3^n + 4^n$. The total model count is then $(3^n + 4^n)^n$.

Counting The most impressive improvements are attained when propositional model counters run in time *exponential* in n, yet first-order model counters run in *polynomial* time. To consider an example where this comes up, let Δ be

$$\forall x, \forall y,\ \texttt{Smokes}(x) \land \texttt{Friends}(x, y) \Rightarrow \texttt{Smokes}(y). \tag{6}$$

This time, the clauses in the grounding of Δ are no longer independent, and it would be wrong to exponentiate. Let us first assume we know that k people smoke, and that we know their identities. Then, how many models are there? Formula 6 encodes that a smoker cannot be friends with a non-smoker. Hence, out of n^2 $\texttt{Friends}$ atoms, $k(n-k)$

have to be false, and the others can take either truth value. Thus, the number of models is $2^{n^2-k(n-k)}$. Second, we know that there are $\binom{n}{k}$ ways to choose k smokers, and k can range from 0 to n. This results in the total model count of $\sum_{k=0}^{n}\binom{n}{k}2^{n^2-k(n-k)}$. WFOMC solvers can automatically construct this formula and compute the model count in time polynomial in n. On the other hand, propositional algorithms require time exponential in n.

Skolemization The theories have so far consisted of universally quantified clauses. Directed models and probabilistic programs also require existential quantification. *Skolemization* is the procedure of eliminating existential quantifiers from a theory. Van den Broeck, Meert, and Darwiche (2013) introduce a Skolemization procedure that is sound for the WFOMC task. Suppose that we are eliminating the existential quantifier in the following sentence from Theory 2.

$$\forall p, \exists c, \mathtt{Card}(p, c)$$

We can do so without changing the model count. First, introduce a new relation $\mathtt{S}$ and replace the sentence by

$$\forall p, \forall c, \mathtt{Card}(p, c) \Rightarrow \mathtt{S}(p)$$

Second, extend the weight function w with $\mathrm{w}(\mathtt{S}(\mathbf{y})) = 1$ and $\mathrm{w}(\neg\mathtt{S}(\mathbf{y})) = -1$ for all $\mathbf{y}$.

We can verify this transformation as follows. For a fixed position p, consider two cases: $\exists c, \mathtt{Card}(p, c)$ is either true or false. If it is true, then so is $\mathtt{S}(p)$. All models of the Skolemized sentence are also models of the original sentence, and the models have the same weight. If $\exists c, \mathtt{Card}(p, c)$ is false, then this does not correspond to any model of the original sentence. However, the Skolemized sentence is satisfied, and $\mathtt{S}(p)$ can be true or false. Yet, for every model with weight w where $\mathtt{S}(p)$ is true, there is also a model with weight $-w$ where $\mathtt{S}(p)$ is false. These weights cancel out in the WFOMC, and the transformation is sound.

Playing Cards Revisited Let us reconsider the playing card problem from Theory 2. Skolemization yields

$$\forall p, \forall c, \mathtt{Card}(p, c) \Rightarrow \mathtt{S}_1(p)$$
$$\forall p, \forall c, \mathtt{Card}(p, c) \Rightarrow \mathtt{S}_2(c)$$
$$\forall p, \forall c, \forall c', \neg\mathtt{Card}(p, c) \lor \neg\mathtt{Card}(p, c') \lor c = c'$$

One can now show that the WFOMC inference rules are capable of simplifying this theory recursively, and efficiently computing its WFOMC. First, one can apply the counting rule to remove $\mathtt{S}_1$, $\mathtt{S}_2$ and the first two sentences. The groundings of the third sentence are independent for different p, enabling exponentiation of the simplified theory

$$\forall c, \forall c', \neg\mathtt{Card}(c) \lor \neg\mathtt{Card}(c') \lor c = c'.$$

Continued application of the WFOMC inference rules yields the following expression for the model count

$$\#SAT = \sum_{k=0}^{n}\binom{n}{k}\sum_{l=0}^{n}\binom{n}{l}(l+1)^k(-1)^{2n-k-l} = n!$$

This expression is evaluated in time polynomial in the number of cards n. It shows that lifted inference is strictly more powerful than propositional reasoning. The key difference is that WFOMC can assume certain symmetries that propositional techniques cannot take into account.

References

Ahmadi, B.; Kersting, K.; and Natarajan, S. 2012. Lifted online training of relational models with stochastic gradient methods. In *Proceedings of ECML PKDD*.

Beame, P.; Li, J.; Roy, S.; and Suciu, D. 2013. Model counting of query expressions: Limitations of propositional methods. In *Proceedings of ICDT*.

Bui, H. H.; Huynh, T. N.; and Sontag, D. 2014. Lifted tree-reweighted variational inference. In *Proceedings of UAI*.

Chavira, M., and Darwiche, A. 2008. On probabilistic inference by weighted model counting. *Artificial Intelligence*.

Darwiche, A. 2009. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press.

Fierens, D.; Van den Broeck, G.; Bruynooghe, M.; and De Raedt, L. 2012. Constraints for probabilistic logic programming. In *Proceedings of the NIPS Probabilistic Programming Workshop*.

Gogate, V., and Domingos, P. 2011. Probabilistic theorem proving. In *Proceedings of UAI*, 256–265.

Gribkoff, E.; Suciu, D.; and Van den Broeck, G. 2014. Lifted probabilistic inference: A guide for the database researcher. *Bulletin of the Technical Committee on Data Engineering*.

Milch, B.; Marthi, B.; Russell, S.; Sontag, D.; Ong, D. L.; and Kolobov, A. 2007. Blog: Probabilistic models with unknown objects. *Statistical relational learning* 373.

Mladenov, M.; Globerson, A.; and Kersting, K. 2014. Lifted message passing as reparametrization of graphical models. In *UAI*.

Niepert, M., and Van den Broeck, G. 2014. Tractability through exchangeability: A new perspective on efficient probabilistic inference. *Proceedings of AAAI*.

Niepert, M. 2013. Symmetry-aware marginal density estimation. In *Proceedings of AAAI*.

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

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

Singla, P.; Nath, A.; and Domingos, P. 2014. Approximate lifting techniques for belief propagation. In *Proceedings of AAAI*.

Thompson, W. C., and Schumann, E. L. 1987. Interpretation of statistical evidence in criminal trials: The prosecutor's fallacy and the defense attorney's fallacy. *Law and Human Behavior* 11(3):167.

Valiant, L. G. 1979. The complexity of enumeration and reliability problems. *SIAM Journal on Computing* 8(3):410–421.

Van den Broeck, G., and Darwiche, A. 2013. On the complexity and approximation of binary evidence in lifted inference. In *Advances in Neural Information Processing Systems 26 (NIPS)*.

Van den Broeck, G., and Niepert, M. 2015. Lifted probabilistic inference for asymmetric graphical models. In *Proceedings of AAAI*.

Van den Broeck, G.; Taghipour, N.; Meert, W.; Davis, J.; and De Raedt, L. 2011. Lifted Probabilistic Inference by First-Order Knowledge Compilation. In *Proceedings of IJCAI*, 2178–2185.

Van den Broeck, G.; Meert, W.; and Darwiche, A. 2013. Skolemization for weighted first-order model counting. *Proceedings of UAI*.

Van den Broeck, G. 2013. *Lifted Inference and Learning in Statistical Relational Models*. Ph.D. Dissertation, KU Leuven.

Van Haaren, J.; Van den Broeck, G.; Meert, W.; and Davis, J. 2014. Tractable learning of liftable Markov logic networks. In *LTPM*.

Venugopal, D., and Gogate, V. 2014. Evidence-based clustering for scalable inference in markov logic. In *ECML PKDD*, 258–273.