Completeness Results for Lifted Variable Elimination: Appendix

Nima Taghipour **Daan Fierens** **Guy Van den Broeck** **Jesse Davis** **Hendrik Blockeel**
Department of Computer Science, KU Leuven, Belgium

Abstract

In this document, we present proofs for Theorem 2 and 3 (given in the paper), and provide more explanation for the empirical evaluation. Further, we present a procedure for tramsforming weighted model counting (WMC) models to parfactor models.

1 PROOF OF THEOREM 2

Let us first recall the theorem.

Theorem 2 *C-FOVE$^+$ is a complete domain-lifted algorithm for the class of models in which each atom has at most 1 logvar.*

Proof sketch. The proof builds on the proof of Theorem 2 (given in the paper). Note that the approach used in Steps 2 and 3 of the proof of Theorem 2 is also applicable here. The operations in Step 2, which together eliminate the 1-logvar atoms, do not depend on the total number of logvars in the parfactors. Using this approach, we can eliminate all the 1-logvar atoms in any model whose atoms contain at most one logvar. The resulting model can be solved as in Step 3. As was shown in the proof of Theorem 2, the time-complexity of these steps is polynomial in the domain size. The inference procedure is thus domain-lifted. □

2 PROOF OF THEOREM 3

Let us first recall the theorem.

Theorem 3 *Lifted sum-out with the group inversion operator is sound, i.e., it is equivalent to summing out the randvars on the ground level.*

Appearing in Proceedings of the 16^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2013, Scottsdale, AZ, USA. Volume 31 of JMLR: W&CP 31.

We prove the theorem by showing that the corresponding ground operations are both *independent* and *isomorphic.*

Independence. We require the following definition.

Given a set of factors F and set of randvars R, we call a subset of factors $F' \subseteq F$ *mutually closed* with respect to a group of randvars $R' \subseteq R$, if (i) no factor in $F \setminus F'$ contains a randvar $r' \in R'$, (ii) no randvar in $R \setminus R'$ appears in a factor $f' \in F'$, and (iii) each randvar $r' \in R'$ appears in some factor $f' \in F'$.

Now, we show that we can form mutually closed sets of randvars and factors in $R = RV(A_i|C)$ and $F = gr(g)$ by partitioning them into sets in which all elements are permutations of each other (can be derived from one another by a permutation of constants). The set of permutations that defines the partitioning is the minimal permutation group $[\Lambda]$.

Given a set of permutations Λ on $\mathbf{X}$, we define two substitutions θ_1, θ_2 to be in the relation $\sim_\Lambda$ iff $\lambda(\theta_1) = \theta_2$ for some $\lambda \in \Lambda$. Using this relation we can define a partitioning of a set of substitutions Θ as $\mathbf{\Theta}_\Lambda$, where θ and θ' are in the same group if and only if $\theta \sim_\Lambda \theta'$.

As shown in steps 1 and 2 of the operator, for any two factors $g\theta$ and $g\theta'$ that share a randvar from the set $RV(A_i)$, we have $\theta = \lambda(\theta')$, for some $\lambda \in [\Lambda]$. Thus for any $\Theta_i \in \mathbf{\Theta}_{[\Lambda]}$, the set of factors $F_i = \{g\theta | \theta \in \Theta_i\}$ are mutually closed w.r.t. the set of randvars $R_i = \{A_i\theta | \theta \in \Theta_i\}$. This shows that we can divide the problem of summing out $RV(A_i)$ from $gr(g)$ into *independent* problems of summing out each set of randvars R_i from the set of factors F_i.

Isomorphism. We show that the sum-out problems are also *isomorphic*, by a mapping between the ground substitutions that produce ground factors in each group.

To show the isomorphism between groups of $gr(g)$, we note that each group is formed from the factors $\{g\theta | \theta \in \Theta_i\}$, where Θ_i is a group in $\mathbf{\Theta}_{[\Lambda]}$. The one-to-one mapping between the factors can thus be established by a one-to-one mapping between the constants of the grounding substitutions in different groups Θ_i

and Θ_j. This is done by starting from an arbitrary pair of substitutions $\theta_i \in \Theta_i$ and $\theta_j \in \Theta_j$ and mapping the constants that are assigned to the same logvar to each other. It follows then that each substitution $\theta'_i \in \Theta_i$ such that $\lambda(\theta_i) = \theta'_i$ is mapped to exactly one substitution $\theta'_j \in \Theta_j$ such that $\lambda(\theta_j) = \theta'_j$. As such the set of factors (and the set of randvars) are isomorphic up to a renaming of the constants in each group.

This shows that the sum-out problems in different groups are independent and isomorphic. Hence, it is correct to replace them by a single lifted operation, i.e. to solve one instance of the problem for a representative group and generalize the result for all, as is performed in lifted sum-out by the group inversion operator.

3 EXPLANATION ABOUT THE EMPIRICAL EVALUATION

In this section we show how C-FOVE$^+$ solves each of the models used in our empirical evaluation, and compare the complexity of inference in each model.

3.1 The friends and smokers model

This model consists of the following two parfactors (in normal form):

$$g_1 = \phi_1(S(X), F(X,Y), S(Y))|X \neq Y$$
$$g_2 = \phi_2(F(X,Y), F(Y,X))|X \neq Y$$

We first eliminate the **2-logvar** F atoms, as follows. We multiply g_1 and g_2 to compute the product

$$g = \phi(S(X), F(X,Y), F(Y,X), S(Y))|X \neq Y$$

Then we eliminate the F atoms by group-inversion, which results in the parfactor

$$g' = \phi'(S(X), S(Y))|X \neq Y$$

Next, we eliminate the **1-logvar** S atoms as follows. By just-different counting conversion, we rewrite g' as

$$g'' = \phi''(\#_X[S(X)])$$

We then eliminate the S randvars by summing-out the counting formula $\gamma = \#_X[S(X)]$ from g''. The result is a potential with no arguments (a constant). This concludes inference.

Complexity. The most expensive step here, is the elimination of the counting formula $\#_X[S(X)]$, whose range size is $O(n)$, with n the domain size of the logvars. As such the whole process runs in time *linear* in the domain size.

3.2 The collective classification model

Below we abbreviate *Link* to L, and *Class* to C. The model consists of the following parfactors:

$$\forall i,j \in \{1,2\}:$$
$$g_{ij} = \phi_{ij}(C_i(P_1), L(P_1,P_2), C_j(P_2)) \; |P_1 \neq P_2$$
$$g_2 = \phi_2(L(P_1,P_2), L(P_2,P_1))|P_1 \neq P_2$$

Inference in this model follows the same steps as the *friends and smokers* model.

We first eliminate the **2-logvar** L atoms, as follows. We multiply all the 5 parfactors g_2, g_{11}, g_{12}, g_{21} and g_{22} to compute the product g:

$$\phi(C_1(P_1), C_2(P_1), L(P_1,P_2), L(P_2,P_1), C_1(P_2), C_2(P_2)) \; |P_1 \neq P_2$$

Then we eliminate the L atoms by group-inversion, which results in the parfactor

$$g' = \phi'(C_1(P_1), C_2(P_1), C_1(P_2), C_2(P_2))|P_1 \neq P_2$$

Next, we eliminate the **1-logvar** atoms C_1, C_2 as follows. By joint conversion on C_1 and C_2, we rewrite each of their occurrences as a joint atom J_{12}

$$\phi''(J_{12}(P_1), J_{12}(P_1), J_{12}(P_2), J_{12}(P_2))|P_1 \neq P_2,$$

which after a simplification of recurring atoms becomes:

$$g' = \phi''(J_{12}(P_1), J_{12}(P_2)) \; |P_1 \neq P_2,$$

Then, by just-different counting conversion, we rewrite g' as

$$g'' = \phi'''(\#_P[\, J_{12}(P) \,] \,)$$

Finally, we eliminate the C_1 and C_2 randvars by summing-out the counting formula $\#_P[\, J_{12}(P) \,]$ from g''. The result is a potential with no arguments (a constant). This concludes inference.

Complexity. The most expensive step in this process is the elimination of the counting formula $\#_P[\, J_{12}(P) \,]$, whose range size is $O(n^{|range(J_{12})|-1})$, with n the domain size of the logvars. Note that here $|range(J_{12})| = 4$, since $range(J_{12}) = range(C_1) \times range(C_2)$. Thus the whole process runs in time $O(n^3)$, i.e., complexity of inference is *cubic* in the domain size.

3.3 Comparison

Comparing the complexity of inference on the two models, we can explain the difference between the runtime of LVE on each model. The reason for this difference can be traced back to the number of distinct unary predicates in each model. The presence of two distinct unary atoms C_1 and C_2 in the collective classification model, result in cubic complexity, while the friends and smokers model has linear complexity, since it contains only one unary predicate S.

Note that these complexities follow from our analysis in the proof of Theorem 1. We showed in our proof that elimination of 1-logvar atoms can have $O(n^r)$ complexity where r is the largest range size among the (joint) unary atoms. In a model with k unary atoms, r can be $O(2^k)$, since in the worst case we might need to make a joint atom out of all unary atoms. Thus the complexity of lifted inference in such a model is $O(n^{2^k-1})$. In our experiments, $k = 1$ for the *friends and smokers* model, and $k = 2$ for the *collective classification* model.

4 TRANSFORMATION FROM WMC TO PARFACTOR MODELS

In this section we introduce a method for transforming any weighted model counting (WMC) model [6, 5, 2] to an equivalent parfactor model [3, 4, 1], i.e., a transformation from the representation used by WFOMC to the representation used by LVE.

A WMC model $M = (\mathcal{C}, w)$ consists of a set of constrained clauses $\mathcal{C}$ and a weight function w that maps each predicate P to a *weight* $w(P)$. We present a transformation from such a model to an equivalent parfactor model. Given any k-WFOMC model (with clauses containing up to k logvars), the following transformation method returns an equivalent k-logvar parfactor model (with parfactors containing up to k logvars).

Consider a WMC model M with the weighting function w and the set of constrained clauses $\mathcal{C} = \{(Cl_i, C_i)\}_{i=1}^n$, where Cl_i is a disjunction of literals of the form $P(\mathbf{X})$ or $\neg P(\mathbf{X})$, and C_i is a constraint on the logvars. We transform this model to a parfactor model M' consisting of two groups of parfactors:

Weight parfactors First we consider the weight function w. For each predicate P in M we add a parfactor $\phi_P(P(\mathbf{X}))$ to M', with potential ϕ_P defined as: $\phi_P(true) = w(P)$ and $\phi_P(false) = 1 - w(P)$.

Clause parfactors Now we consider the set of constrained clauses $\mathcal{C}$. For each constrained clause $(Cl_i, C_i) \in \mathcal{C}$, we add a parfactor $\phi_i(\mathcal{A}_i)|C_i$ to M', where $\mathcal{A}_i$ is the set of atoms that appear (in negated form) in clause Cl_i, and the potential ϕ_i is defined such that for any assignment of values $\mathbf{a}$ to $\mathcal{A}_i$: $\phi_i(\mathbf{a}) = 1$ if $\mathbf{a}$ satisfies Cl_i, and $\phi_i(\mathbf{a}) = 0$ otherwise.

This transformation maps any WMC model M to a parfactor model M' that defines the same probability distribution as M. The following example illustrates such a transformation.

Example. Consider the 2-logvar WMC model M consisting of the weight function w, and the constrained clause,

$$\neg P(X) \vee Q(Y)|X \neq Y$$

Using the above method we derive the equivalent parfactor model M' consisting of the following set of parfactors:

- Two *weight* parfactors $\phi_P(P(X))$ and $\phi_Q(Q(X))$, with potentials ϕ_P and ϕ_Q defined as follows:

P	ϕ_P
$false$	$1 - w(P)$
$true$	$w(P)$

Q	ϕ_Q
$false$	$1 - w(Q)$
$true$	$w(Q)$

- One *clause* parfactor $\phi(P(X), Q(Y))|X \neq Y$, with potential function ϕ defined as follows:

P	Q	ϕ
$false$	$false$	1
$false$	$true$	1
$true$	$false$	0
$true$	$true$	1

Note that the parfactor model M', similar to the WMC model M, is a 2-logvar model. $\square$

Given any WMC model M, this transformation maps each clause in M to a parfactor that involves the same atoms, in the resulting parfactor model M'. As such, each clause is mapped to a parfactor with the same (number of) logvars. This transformation thus maps any k-WFOMC model into an equivalent k-logvar parfactor model.

References

[1] Rodrigo de Salvo Braz. *Lifted First-order Probabilistic Inference.* PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 2007.

[2] Vibhav Gogate and Pedro Domingos. Probabilistic theorem proving. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 256–265, 2011.

[3] Brian Milch, Luke S. Zettlemoyer, Kristian Kersting, Michael Haimes, and Leslie Pack Kaelbling. Lifted probabilistic inference with counting formulas. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 1062–1608, 2008.

[4] David Poole. First-order probabilistic inference. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 985–991, 2003.

[5] Guy Van den Broeck. On the completeness of first-order knowledge compilation for lifted probabilistic inference. In *Proceedings of the 24th Annual Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 1386–1394, 2011.

[6] Guy Van den Broeck, Nima Taghipour, Wannes Meert, Jesse Davis, and Luc De Raedt. Lifted probabilistic inference by first-order knowledge compilation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2178–2185, 2011.