From Lifted Inference to Lifted Models

Daan Fierens
Department of Computer Science
KULeuven, Belgium
daan.fierens@cs.kuleuven.be

Kristian Kersting
University of Bonn and Fraunhofer IAIS, Germany
kristian.kersting@iais.fraunhofer.de

Abstract

In this position paper we raise the question whether lifted inference can be performed by 'lifting' the probabilistic model rather than the inference algorithm. We show that this is indeed possible using so-called count models — at least for the common examples used within the exact lifting literature. The challenge is to turn this into a general approach.

1 Lifted Probabilistic Models

Today, lifted probabilistic inference applies a 'lifting' algorithm to a given model (e.g. [1, 4, 5]). Here, we ask the question whether one could instead compile the given model to another model — the 'lifted' model — such that standard propositional inference on the lifted model corresponds to (or is as efficient as) lifted inference on the original model. This has recently been done for lifted solving of linear programs and inference with pairwise Gaussian MRFs [6], but not yet for probabilistic models in general. Such lifted models help to characterize what lifted inference really is. E.g., they naturally suggest to view lifted inference as efficient inference in an 'over-sized' ground model that includes 'redundant' variables with shared parameters. The 'lifted model' approach compiles the 'over-sized' into a 'just-right sized' model. It also takes away the burden of developing lifted versions of propositional inference algorithms.

What could a 'lifted model' look like? We propose to compile a given model to a *count model*: a propositional model defined over count random variables rather than the original variables.

2 Count Models by Example

We consider the task of computing marginals or max-marginals in factor graphs defined over Boolean random variables. Our compilation approach consists of three steps: **(S1)** construct a count model, **(S2)** perform inference on the count model using any propositional algorithm, and **(S3)** 'recover' the marginals. Fig. 1 illustrates this approach for models commonly used in the exact lifted inference literature [1, 2, 4, 5]. Below we explain the steps for computing marginals in the `sick-death` example [1], Fig. 1**(a,b)**.

(S1a) Structure of the Count Model. Given a factor graph (be it in parfactor or ground form), we first determine the structure of the count model. The idea is that variables for which (the computations of) the marginals are identical are 'collapsed' into a single *count variable*. A count variable that covers a group of variables g in the original model represents the number of variables in g having value true (and hence has $|g|+1$ possible values). E.g., all variables $Sick(p)$ are collapsed into a count variable $\#Sick$. When given the ground factor graph, one way to find clusters of variables is to run color passing [4]. For the examples reported here, this gives the appropriate result. In general, however, color passing can *over*cluster (it clusters variables indistinguishable *for BP*), yielding a count model that can only give approximate marginals.

(S1b) Parameters of the Count Model. Given the structure of the count model, we compute its potentials. For `sick-death`, we need two factors: $\phi(E, \#S)$ (to be derived from the factor $\phi(E, S(p))$ in the original model) and $\phi(\#S, \#D)$ (from $\phi(S(p), D(p))$. The corresponding equations are given in Fig. 1**(a)** and illustrated for domain size 3 in Fig. 1**(b)**. As an example, the entry $\phi(\#S=1, \#D=1)$ in Fig. 1**(b)** can be interpreted as follows: it covers all scenarios with 1 sick and 1 dead person. One scenario is that the same person is sick and dead, so 2 other people are non-sick and alive, yielding a contribution of $\beta_{ff}^2\beta_{tt}$ on the ground level, to be multiplied with 3 (multinomial coefficient) because there are 3 choices for the sick&dead person. The other scenario is that the sick and the dead person are different, yielding $\beta_{ff}\beta_{ft}\beta_{tf}$, to be multiplied by 6.

(One might be tempted to include bi- or multinomial coefficients also in the other factor $\phi(E, \#S)$, but that is incorrect because it double-counts scenarios.) Note that the equations for the potentials in Fig. 1**(a)** are expressed as function of the domain size N. Hence our count model is a fully *just-right parameterized* model, applicable for any domain size.

(S2) Inference. The count model is a regular factor graph, so we can use any propositional inference algorithm (in practice, the large number of states of a count variable can be a burden). For all examples shown, exact inference on the count model yields (after **S3**) exact marginals for the original model.
Before moving to the final step (**S3**), let us compare the characteristics of the count model and the original ground model. On the negative side, the *size* (number of entries) of the factors in the count model is larger than for the original model (e.g. for `sick-death`). On the positive side, however, even when the original model is *loopy*, the count model can be a tree, allowing for efficient exact inference (e.g. Fig. 1**(d–f)**). More generally, the count model typically has lower *treewidth* than the original model. Also, the count model sometimes has a lower *order* (max. number of variables per factor) than the original model. This happens e.g. for `friends&smokers`, where the original model has order 3 but the count model has order 2, see Fig. 1**(f)**. This is relevant when computing max-marginals since many such algorithms (e.g. LP-based ones) assume a pairwise model (i.e. order 2).

(S3) Recovery. Inference on the count model yields the marginals for E, $\#S$ and $\#D$. From the marginals of $\#S$ (which is a count variable with $N + 1$ possible values), we can derive the marginals of the original $S(p)$ variables (which are identical for all p) by applying the *recovery equations*, see Fig. 1**(c)**, and normalizing the results. We can do the same for $\#D$ and $D(p)$. The recovery equations always have the same form as in Fig. 1**(c)**, regardless of the model considered.

Discussion. For all examples in Fig. 1, the lifted model yields the same exact marginals as the original ground model (if, of course, we perform exact inference in **S2**). All steps take time at most polynomial in the domain size N, so the approach is 'lifted' according to Van den Broeck's definition [2]. The fact that these original models with all the individual variables can be converted into count models involving only the count variables suggests that the original models are in some sense 'misspecified' or 'oversized', and that the conversion can be seen as re-sizing the model (see also the work on lifting linear programs [6]). The idea of having random variables that represent counts is related to other work in lifted inference like C-FOVE's *counting conversion* [5]. Although C-FOVE also works with count variables ('counting formulas'), it never entirely replaces the group of individiual variables by the corresponding count variable yielding a purely propositional model. This also surfaces in the fact that C-FOVE has no counterpart of our recovery equations. Another difference between lifted models and (as far as we know) all *exact* lifted inference approaches is that they only compute the marginals for one (cluster of) variable(s), while we can compute all marginals in parallel using an appropriate inference algorithm in **S2**.

The Max-Marginal Case. The above is for computing marginals. Max-marginals require a slightly different count model. For the equations defining the potentials, Fig. 1**(a)**, as well as the recovery equations, Fig. 1**(c)**, the max-marginal form can be obtained from the given equations by (1) replacing all sums by maximizations and (2) omitting all coefficients (like $C(N, i, s, d)$ in Fig. 1**(a)**, s/N in Fig. 1**(c)**, etc.).

3 Outlook and Challenges

While we have a conversion to a count model for most examples from the literature on exact lifting, we currently do not have a fully general conversion method, in particular for deriving the potentials of the count model (**S1b**). Yet, it is clear that many models contain reoccurring 'patterns'. Some simple patterns of dependencies are: *one-to-one* (e.g. the factor between $Sick(p)$ and $Death(p)$ in `sick-death`, but also in `friends&smokers` between $Smokes(p)$ and $Cancer(p)$), *one-to-many* (e.g. between $Epidemic$ and $Sick(p)$, but also in `workshop-attributes` and `competing-workshops`), and *many-to-many* (e.g. in `competing-workshops`). Each pattern requires a specific way of defining the corresponding potential in the count model. Of course there are many other, more complex, patterns possible, e.g. with *self-loops* as in `friends&smokers`. Ideally, we need a set of general conversion rules (like lifted compilation rules [2]). We currently do not have such a general treatment, and even suspect that the conversion is not always possible. Rather, the question seems to be for *which classes* of models it is possible. Another interesting question is whether we can derive closed-form equations for the marginals, parameterized by the domain size (as sometimes done in 'counting programs' [3]). Our conversion to count models might be a step in this direction. The count model itself is already parameterized by the domain size (e.g. the potentials in Fig. 1**(a)**). For all examples shown, the count model is a small tree or chain, and deriving the resulting marginals in closed form can be done.

We believe that further working out the idea of lifted models is a promising but also challenging future avenue for StarAI.

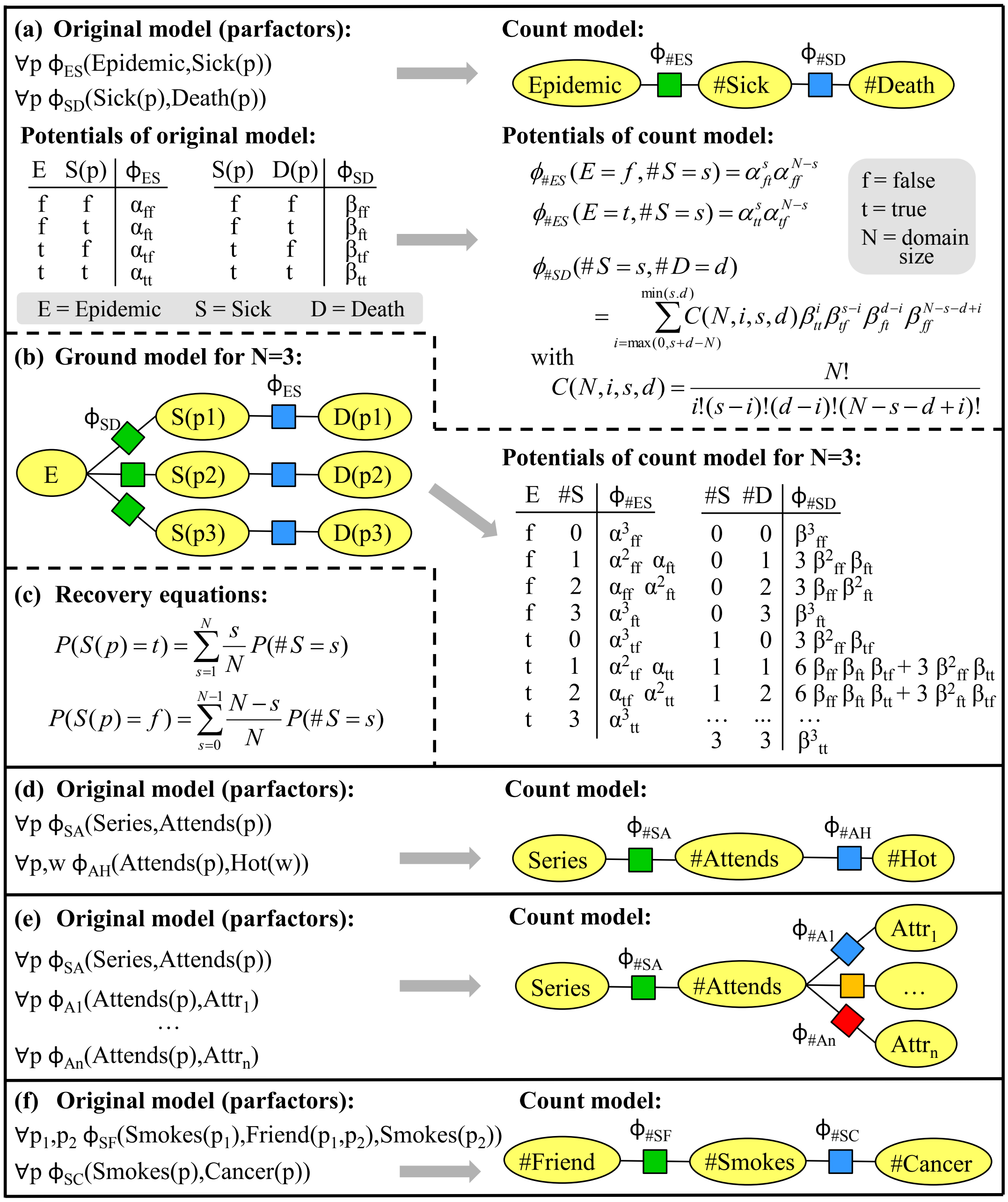

Figure 1: Examples of the conversion to a count model: **(a)-(c)** Sick death (detailed example), **(d)** Competing workshops, **(e)** Workshop attributes, **(f)** Friends & smokers.

Acknowledgements. Daan Fierens is a post-doctoral fellow of the Research Foundation-Flanders (FWO-Vlaanderen). Kristian Kersting is supported by the Fraunhofer ATTRACT fellowship STREAM and by the EC (FP7-248258-First-MM).

References

[1] R. de Salvo Braz, E. Amir, and D. Roth. Lifted first-order probabilistic inference. In IJCAI 2005.

[2] G. Van den Broeck. On the completeness of first-order knowledge compilation for lifted probabilistic inference. In NIPS 2011.

[3] A. Jha, V. Gogate, A. Meliou, and D. Suciu. Lifted inference seen from the other side: The tractable features. In NIPS 2010.

[4] K. Kersting, B. Ahmadi, and S. Natarajan. Counting belief propagation. In UAI 2009.

[5] B. Milch, L. S. Zettlemoyer, K. Kersting, M. Haimes, and L. P. Kaelbling. Lifted probabilistic inference with counting formulas. In AAAI 2008.

[6] M. Mladenov, B. Ahmadi, and K. Kersting. Lifted linear programming. In AISTATS 2012.