Reasoning about Large Populations with Lifted Probabilistic Inference

Kristian Kersting, Brian Milch, Luke S. Zettlemoyer, Michael Haimes, Leslie Pack Kaelbling
MIT CSAIL, Cambridge, MA 02139
{kersting,milch,lsz,mhaimes,lpk}@csail.mit.edu

Abstract

We use a concrete problem in the context of planning meetings to show how lifted probabilistic inference can dramatically speed up reasoning. We also extend lifted inference to deal with cardinality potentials, and examine how to deal with background knowledge about a social network.

Lifted inference: An example. Suppose that n people (say, $n = 100$) have been invited to a NIPS workshop, and we are wondering whether the attendees will overflow the 40-seat room we have reserved. A graphical model for this scenario is shown in Fig. 1(a). In this simple model, the attendance variables $\texttt{attend}(p_i)$ for each person p_i are conditionally independent given the workshop's popularity. We get noisy information about each person's attendance based on the reply they have sent us: "yes", "no", or "no reply".

Assume for the moment that we just want to estimate the workshop's popularity (ignoring the `roomOverflow` variable for now). In this case, we need to compute the marginal distribution for the `popularity` random variable given the `reply` variables. One commonly used algorithm, variable elimination (VE), computes this marginal by eliminating each $\texttt{attend}(p_i)$ variable in turn: for each p_i, it first multiplies together the factors $\phi(\texttt{attend}(p_i), \texttt{reply}(p_i))$ and $\phi(\texttt{attend}(p_i), \texttt{popularity})$, then sums out $\texttt{attend}(p_i)$ to get a factor on `popularity` alone. The resulting n factors on `popularity` are then multiplied together and normalized to yield a posterior distribution. The time required is linear in n.

The basic insight of *lifted* inference algorithms such as *first-order VE* (FOVE) [3, 1] is that because this model treats the n invitees interchangeably, VE ends up doing the same multiplications and summations over and over. We can avoid this repeated work if we explicitly represent the interchangeability of entities. Fig. 1(b) shows how this is done. Instead of specifying factors for each person separately, we use *parameterized factors* or *parfactors*, where the random variables involved in the factor are parameterized by logical variables. In our case, we need just two parfactors $\phi(\texttt{attend(P)}, \texttt{reply(P)})$ and $\phi(\texttt{attend(P)}, \texttt{popularity})$, which apply to all people `P`. Given observations about people's replies, the FOVE algorithm *shatters* each of these parfactors into three copies, one for the n_+ people who said "yes", one for the n_- who said "no", and one for the n_0 people who did not reply. It then performs elimination just three times, once for each of these groups, rather than n times as before. This yields three factors on `popularity`, which we will call ϕ_+, ϕ_-, and ϕ_0. The posterior distribution on `popularity` is now proportional to $\phi_+(\texttt{popularity})^{n_+} \times \phi_-(\texttt{popularity})^{n_-} \times \phi_0(\texttt{popularity})^{n_0}$. Assuming unit cost for exponentiation, this lifted algorithm takes constant time, removing the linear dependence on n.

Cardinality potentials. Now consider our original goal of predicting whether the attendees will overflow a 40-seat room. For this purpose, we can query the `roomOverflow` variable in Fig. 1, which deterministically indicates whether more than 40 $\texttt{attend}(p_i)$ variables are true. The first difficulty here is that a tabular representation of the factor linking `roomOverflow` to all the $\texttt{attend}(p_i)$ variables would require space exponential in n. Although more compact representations for such factors have been developed [4], the FOVE algorithm [1] does not exploit them.

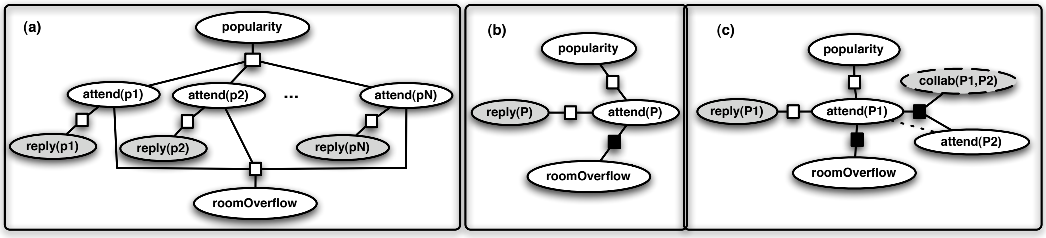

Figure 1: *Meeting* Markov networks using **(a)** standard factors and **(b,c)** parfactors. Ovals denote random variables (shaded if observable). White boxes denote standard parfactors whereas black boxes encode parfactors with cardinality potentials. In contrast to **(b)**, **(c)** also encodes the social network among invitees using the deterministic (shaded, dashed oval) relation $\texttt{collab}(\texttt{P1}, \texttt{P2})$. The dotted edge between $\texttt{attend}(\texttt{P1})$ and $\texttt{attend}(\texttt{P2})$ indicates that they refer to the same set of random variables.

We extend FOVE to take advantage of what Gupta *et al.* [2] have called *cardinality potentials*. We represent these potentials with notation such as $\phi(\texttt{roomOverflow}, \#_{\texttt{P}}[\texttt{attend}(\texttt{P})])$, indicating that the value of the potential depends only on *how many* of the $\texttt{attend}(p_i)$ variables have each particular value. This restricted form of dependency can be exploited to speed up inference. For example, when summing out the $\texttt{attend}(p_i)$ variables in our model, we do not need to iterate over their 2^n possible instantiations: it suffices to iterate over all possible *histograms* assigning counts to `true` and `false` that add up to n (actually, we need to do three separate iterations over histograms, one for each group of people with the same reply). The number of such histograms is only linear in n, so we have an exponential speed-up.

This is the same insight exploited by *counting elimination* in the existing version of FOVE [1]. However, we extend FOVE by allowing cardinality potentials as input. We also allow cardinality potentials to be stored as intermediate results of the elimination process, which extends the set of cases where inference can be performed at a lifted level.

Social networks. In reality, people's decisions about whether to attend a workshop depend on how many of their friends or collaborators are attending. Fig. 1(c) shows a model where the `attend` variable for each person `P1` is linked to the `attend` variables for `P1`'s neighbors in a known social network, represented by the relation $\texttt{collab}(\texttt{P1}, \texttt{P2})$. In our extended version of FOVE, we can represent this linkage using a cardinality potential with $\texttt{collab}(\texttt{P1}, \texttt{P2})$ as a constraint: $\phi(\texttt{attend}(\texttt{P1}), \#_{(\texttt{P2}:\texttt{collab}(\texttt{P1},\texttt{P2}))}[\texttt{attend}(\texttt{P2})])$.

The social network creates distinctions among the invitees, making lifted inference more difficult; indeed, the existing version of FOVE does not allow relations as constraints [1]. However, lifted inference should be able to avoid repeated work when the network has many small connected components — such as research groups or families — that are isomorphic to each other. In other cases, it may be possible to approximate the network using structures that are amenable to lifted inference.

Conclusions. With the goal of applying lifted inference "in the wild", we have extended FOVE to deal with *cardinality potentials*. Other aspects of this research program include lifted filtering for temporal models, and approximate lifting methods that treat some entities as interchangeable even when their potentials are not exactly the same.

Acknowledgments. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under Contract No. NBCHD030010.

References

[1] R. de Salvo Braz, E. Amir, and D. Roth. Lifted first-order probabilistic inference. In *Proc. 19th IJCAI*, pages 1319–1325, 2005.

[2] R. Gupta, A.A. Diwan, and S. Sarawagi. Efficient inference with cardinality-based clique potentials. In *Proc. 24th ICML*, 2007.

[3] D. Poole. First-order probabilistic inference. In *Proc. 18th IJCAI*, pages 985–991, 2003.

[4] N. L. Zhang and D. Poole. Exploiting causal independence in Bayesian network inference. *J. Artif. Intell. Res.*, 5:301–328, 1996.