Pairwise Markov Logic

Daan Fierens[1], Kristian Kersting[2], Jian Chen[1], and Jesse Davis[1]

[1] Department of Computer Science, KULeuven, Belgium
{daan.fierens,jesse.davis}@cs.kuleuven.be
[2] University of Bonn and Fraunhofer IAIS, Germany
kristian.kersting@iais.fraunhofer.de

Abstract. For many tasks in fields like computer vision, computational biology and information extraction, popular probabilistic inference methods have been devised mainly for propositional models comprising of unary and pairwise clique potentials. In contrast, statistical relational approaches typically do not restrict a model's representational power and use high-order potentials to capture the rich statistics of relational domains. This paper aims to bring both worlds closer together. We introduce pairwise Markov Logic, a subset of Markov Logic where each formula contains at most two atoms. We show that every non-pairwise Markov Logic Network (MLN) can be transformed or 'reduced' to a pairwise MLN. Thus, existing, highly efficient probabilistic inference methods can be employed for pairwise MLNs without the overhead of devising or implementing high-order variants. Experiments on two relational datasets confirm the usefulness of this reduction approach.

1 Introduction

Low-order models represent one of the most popular ways to model many probabilistic inference tasks in domains like computer vision, computational biology, and information extraction. The *order* of a factor graph is the maximum number of arguments of a factor; the order of a Markov network is the size of the largest clique [1]. A model is called *pairwise* if its order is (at most) two. Based on a connection with statistical physics, it is common to write a pairwise model with unary and pairwise clique potentials, understand the Bayesian priors it incorporates, and then perform inference. Therefore, most state-of-the-art methods for probabilistic inference were mainly developed for such pairwise models. High-order variants of such inference methods often do not exist and if they do, they are typically more complex and lack implementations. This is particularly true for MAP/MPE inference methods based on Linear Programming [2], graph cuts, etc. While most of these methods can in principle work on non-pairwise models, pairwise models receive the most attention because this facilitates implementation and theoretical analysis (e.g., convergence analysis for iterative methods like Belief Propagation [2]).

For Markov Logic Networks (MLNs) [3], the notion of pairwise Markov Logic has not yet been explored.[1] Following the propositional case, we define the *order*

[1] Pairwise variants of other statistical relational models have not been studied either.

of an MLN as the maximum number of atoms in a formula, i.e., the maximum 'length' of a formula in the MLN. We call an MLN (or a single MLN formula) *pairwise* if its order is two. For instance, the formula $Smokes(x) \Rightarrow Asthma(x)$ is pairwise, but $Friends(x,y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$ is not. We call the latter a *triplewise* formula since it has length three.

Pairwise MLNs have advantages for both ground and lifted inference. Ground inference typically applies methods from the graphical models literature, many of which focus on pairwise models. A similar argument holds for lifted inference, where recent work [4, 5] uses graph-theoretical notions that assume that the network is pairwise (as it can then be represented as a simple graph rather than a hypergraph). However, despite the advantages of pairwise MLNs, one should not discard non-pairwise MLNs. It is difficult, if not practically impossible, to capture many of the rich statistics of relational domains using a pairwise model. For example, when performing structure learning, restricting the hypothesis space to pairwise formulas would simply ignore too many relevant patterns in relational datasets. Many typical relational patterns require triplewise formulas. Common examples are found in collective classification (e.g., $Class(x,c) \wedge Link(x,y) \Rightarrow Class(y,c)$), link prediction (e.g., $Property(x) \wedge Property(y) \Rightarrow Link(x,y)$), social networks (e.g., the above Smokes formula), etc.

Therefore, it is common to work with a non-pairwise model during modelling and learning but to transform or to *reduce* the model to an equivalent pairwise model for inference. This paper shows that this can be done for Markov Logic. Specifically, we show that any triplewise MLN (containing formulas of at most length three) can be reduced to a pairwise MLN. While our approach naturally extends beyond triplewise MLNs, we only consider this case as it is sufficient for many relational domains (cf. the above example formulas) and it simplifies the presentation. Experiments on two common relational datasets demonstrate the usefulness of the reduction approach.

Section 2 explains the reduction for propositional MLNs and Section 3 for first-order MLNs. Section 4 presents our experimental results.

2 Reduction for Propositional MLNs

A propositional MLN is a set of pairs (F_i, w_i) where F_i is a propositional logic formula (using connectives $\neg$, $\wedge$, $\vee$, $\Rightarrow$ and $\Leftrightarrow$) and w_i is a real-valued weight. A propositional MLN defines a probability distribution on the set of possible worlds (interpretations): the probability of a world ω is $P(\omega) = \frac{1}{Z} exp(\sum_i w_i \delta_i(\omega))$, with Z a normalization constant and $\delta_i(\omega)$ the indicator function being 1 if formula F_i is true in world ω and 0 otherwise. Since we only consider triplewise MLNs here, every formula F_i has length at most three.

Below we show how to reduce a triplewise MLN to a pairwise MLN. In graphical models, reduction is typically done by converting the model (or its energy function) to a pseudo-Boolean function or multi-linear polynomial [1]. While our reduction can also be phrased in this terminology, we instead chose for a more self-contained formulation, referring only to Markov Logic. We achieve reduction by means of a rewriting process that operates on the MLN knowledge base (the set of weighted formulas). This is a two-step process, the first step is an enabling step, the second step does the actual reduction to a pairwise MLN.

Step 1: Rewrite triplewise formulas to positive normal form

We first rewrite every triplewise formula in the MLN to a set of formulas in what we call *positive normal form*. A formula is in positive normal form if it is a conjunction of atoms (not involving negation). For example, consider the propositional MLN formula $P \wedge Q \Rightarrow R$ with weight w. Rewriting it to positive normal form yields three formulas: $P \wedge Q \wedge R$ with weight w, $P \wedge Q$ with weight $-w$, and *true* with weight w. The equivalence of these three formulas to the original formula can be shown by constructing the table of $2^3 = 8$ possible worlds of the three involved atoms and comparing the resulting distributions. The formula *true* does not change the distribution and can be dropped.

More generally, a triplewise propositional formula F_{orig} with weight w can be rewritten in positive normal form by replacing it by a set of at most seven new formulas: **(a)** the triplewise formula $P \wedge Q \wedge R$; **(b)-(d)** the three pairwise formulas $P \wedge Q$, $P \wedge R$, and $Q \wedge R$; **(e)-(g)** the three unary formulas P, Q, and R. The weights that we need to assign to each of these seven formulas in order to be equivalent to F_{orig} can be calculated from the truth table of F_{orig}. Let α_{pqr} be the Boolean truth value (1 or 0) of formula F_{orig} under the interpretation $P = p$, $Q = q$ and $R = r$ (e.g. α_{ttf} is the truth value of F_{orig} when P and Q are true and R is false). The weights of the seven formulas should be set to: **(a)** $w(\alpha_{ttt} - \alpha_{ttf} - \alpha_{tft} + \alpha_{tff} - \alpha_{ftt} + \alpha_{ftf} + \alpha_{fft} - \alpha_{fff})$, **(b)** $w(\alpha_{ttf} - \alpha_{tff} - \alpha_{ftf} + \alpha_{fff})$, **(c)** $w(\alpha_{tft} - \alpha_{tff} - \alpha_{fft} + \alpha_{fff})$, **(d)** $w(\alpha_{ftt} - \alpha_{ftf} - \alpha_{fft} + \alpha_{fff})$, **(e)** $w(\alpha_{tff} - \alpha_{fff})$, **(f)** $w(\alpha_{ftf} - \alpha_{fff})$, **(g)** $w(\alpha_{fft} - \alpha_{fff})$. The fact that the result is equivalent to the original formula can again be proven by enumerating the 8 possible worlds of P, Q and R and comparing the distributions. Applying these equations on our earlier example ($P \wedge Q \Rightarrow R$) indeed gives the result mentioned earlier ($P \wedge Q \wedge R$ gets weight w, $P \wedge Q$ gets weight $-w$, and the five other formulas get weight 0 and are dropped). This rewriting process is carried out for each triplewise formula in the MLN consecutively.

Step 2: Reduce triplewise formulas to sets of pairwise formulas

After the previous step, the MLN still contains triplewise formulas, but each one is guaranteed to be in positive normal form: $P \wedge Q \wedge R$. Such a formula can then be rewritten or 'reduced' to a set of pairwise formulas by introducing new, *auxiliary*, propositions.

Rewriting a triplewise formula $P \wedge Q \wedge R$ with weight w to a set of pairwise formulas requires one auxiliary proposition, denoted A below. The set of pairwise formulas produced by the reduction depends on the sign of w. If $w > 0$, we need four formulas: **(a)-(c)** the pairwise formulas $A \wedge P$, $A \wedge Q$ and $A \wedge R$, each with weight w, **(d)** the unary formula A, with weight $-2w$. If $w < 0$, we need seven formulas: **(a)-(c)** the pairwise formulas $P \wedge Q$, $P \wedge R$ and $Q \wedge R$, each with weight w, **(d)-(f)** the pairwise formulas $A \wedge P$, $A \wedge Q$ and $A \wedge R$, each with weight $-w$, **(g)** the unary formula A, with weight w. This rewriting process is carried out for each triplewise formula separately, each time with a different auxiliary proposition (i.e., each auxiliary proposition is given a different name to ensure that the reductions of two different triplewise formulas do not become dependent).

The set of pairwise formulas that results from reducing a triplewise formula $P \wedge Q \wedge R$ is *max-equivalent* to the original formula. The original formula defines a distribution $P_{orig}(P,Q,R)$ on the set of possible worlds of the three involved atoms, while the new set of formulas defines a distribution $P_{new}(A,P,Q,R)$ since it includes the auxiliary proposition A. Max-equivalence means $P_{orig}(P,Q,R) = max_A P_{new}(A,P,Q,R)$. In other words: 'maxing-out' the auxiliary atom from the new distribution yields the original distribution.[2] Again, the equivalence can be proven by enumerating the possible worlds and comparing the distributions. This local equivalence also carries over globally: if we reduce an entire triplewise MLN to a pairwise MLN in this way, the pairwise MLN will be max-equivalent to the original MLN, if we max-out all auxiliary atoms.

3 Reduction for First-Order MLNs

So far, we have only considered propositional MLNs. A first-order MLN is a set of pairs (F_i, w_i) where each F_i is a formula in first-order logic (we do not allow existential quantifiers since they often require special treatment in Markov Logic). The resulting probability distribution is: $P(\omega) = \frac{1}{Z} exp(\sum_i w_i n_i(\omega))$, with $n_i(\omega)$ the number of true groundings of formula F_i in world ω.

Reducing a first-order MLN can be done at the ground level or at the first-order / lifted level. Reducing at the ground level simply consists of using existing methods to ground the MLN and then carrying out the reduction as in the previous section, treating each ground atom as a separate proposition (this requires using a separate auxiliary proposition for each grounding of a triplewise formula).

The reduction can also be done at the lifted level. Because of space restrictions we explain this by example.[3] Consider the triplewise formula $Class(x, c_1) \wedge Link(x, y) \Rightarrow Class(y, c_2)$ with weight $w > 0$. We use the MLN convention of writing logical variables ('*logvars*') such as x in lowercase. Step 1 rewrites this formula to positive normal form, yielding two formulas $Class(x, c_1) \wedge Link(x, y) \wedge Class(y, c_2)$ with weight w, and $Class(x, c_1) \wedge Link(x, y)$ with weight $-wN_c$, see below. Step 2 reduces the first of these two formulas to the following four formulas: **(a)-(c)** $A(x, y, c_1, c_2) \wedge Class(x, c_1)$ and $A(x, y, c_1, c_2) \wedge Link(x, y)$ and $A(x, y, c_1, c_2) \wedge Class(y, c_2)$, each with weight w, and **(d)** $A(x, y, c_1, c_2)$, with weight $-2w$. Note the resemblance with our earlier propositional example ($P \wedge Q \Rightarrow R$), but with two complications due to working at the first-order level. First, in Step 1, the weight of the second formula needs to be $-wN_c$ instead of simply $-w$, with N_c the number of classes (the domain size of the type 'class').

[2] Maxing-out is the counterpart of summing-out (marginalization).

[3] *Note to reviewers:* we plan to expand this part if invited for a 12-page submission.

This is because this formula does not have the logvar c_2 that appears in the original formula (i.e., we 'lost' the logvar c_2). Second, the auxiliary predicate A needs as arguments all logvars that appear in the triplewise formula (x, y, c_1 and c_2), to ensure that the reductions for the different groundings are independent.

4 Experimental Evaluation

To evaluate the usefulness of our reduction, we investigate MAP inference tasks. (also called MPE) [3]. That is, we observe the truth value of a subset of all ground atoms and need to find the most likely state of all other ground atoms.[4]

Datasets. We considered two first-order MLNs, each containing one triplewise first-order formula and several pairwise and unary formulas. The first MLN is for the synthetic Smokers domain (the triplewise formula is given in Section 1), the second MLN is for the real-world WebKB dataset (the triplewise formula is the one given in Section 3). For both MLNs, we considered 6 different domain sizes (number of people, respectively webpages). For each domain size, we generated 20 different MAP task instances by randomly selecting the required number of entities in the domain (people/webpages) and then randomly selecting 80% of all resulting ground atoms as evidence, leaving the remaining 20% to be maxed-out. Hence, in total we obtained 120 MAP tasks for each MLN.

Algorithms. We compared two MAP algorithms. The first is MaxWalkSAT [3], which works in the same way irrespective of whether the MLN is pairwise or not. The second is MPLP [2] (a Max Product variant based on Linear Programming), which is most easily formulated for pairwise models. To run MPLP, we reduced the triplewise MLN to a pairwise MLN using the ground-level approach, converted the result to a Markov net, and fed this to MPLP. We refer to this as *MPLP-p* ('p' for pairwise). We also ran MaxWalkSAT (MWS) both on the original triplewise MLN (*MWS-t*) and on the reduced pairwise MLN (*MWS-p*). For each considered MAP task, we first ran MPLP-p and measured the runtime. Since MWS is an anytime algorithm, we then run both MWS-p and MWS-t with five different time-budgets: 0.5, 1, 2, 5 and 10 times the runtime of MPLP-p. We evaluated the quality of the MAP assignment returned by an algorithm by computing the sum of weights of satisfied formulas in the triplewise MLN under this assignment, as this sum is proportional to the probability of the MAP assignment [3].

Summary of results. We found that MPLP-p is far superior to both MWS-t and MWS-p. If we give MPLP-p and MWS-t the same runtime and compare the quality of the returned MAP assignment, the wins/ties/losses on Smokers are 24/90/6, meaning that on 24 tasks MPLP-p is better, on 90 tasks there is a tie, and on 6 tasks MWS-t is better. On WebKB, the results are quite extreme: 119/1/0, so MPLP-p is almost always strictly better than MWS-t. Also when comparing MPLP-p against MWS-p (instead of MWS-t), MPLP-p is superior:

[4] MAP inference maxes-out all unobserved variables. MAP on a pairwise MLN resulting from a reduction of a triplewise MLN hence maxes-out all auxiliary atoms, which is exactly what is needed for having max-equivalence.

with the same amount of runtime, wins/ties/losses are 120/0/0 on Smokers and again 119/1/0 on WebKB. On WebKB, MPLP-p is better than both MWS-t and MWS-p even if we give the latter two 10 times the runtime of MPLP-p.

We conclude that MPLP on the reduced pairwise MLN is superior to MWS on the original triplewise MLN or the reduced pairwise MLN. While MPLP can work with triplewise MLNs, it is significantly easier to formulate and implement on pairwise MLNs. The same applies to other MAP algorithms, like those based on linear programming and graph cuts. Hence, by reducing to a pairwise MLN, we gain easier access to a broader range of algorithms. As our experiments with MPLP show, this can yield significantly better results compared to algorithms, like MWS, that work in the same way regardless of whether the MLN is pairwise. This shows the usefulness of our reduction approach.

5 Conclusion

We introduced Pairwise Markov Logic as a subset of Markov Logic that is of special interest. We have shown how to reduce a non-pairwise MLN to an equivalent pairwise MLN. Working with pairwise MLNs is advantageous because many ground and lifted probabilistic inference method focus on this case. Our experiments with MAP/MPE inference confirm this. While we focussed on inference here, also learning will benefit from this work, as MAP/MPE inference is a subprocedure in many learning tasks (e.g., discriminative weight learning for MLNs [3]). Using pairwise MLNs for lifted inference is interesting future work.

Acknowledgements. DF is a post-doctoral fellow of the Research Foundation-Flanders (FWO-Vlaanderen). KK is supported by the Fraunhofer ATTRACT fellowship STREAM and by the EC (FP7-248258-First-MM). JD is supported by the Research Fund K.U.Leuven (CREA/11/015 and OT/11/051), EU FP7 Marie Curie Career Integration Grant (#294068) and FWO-Vlaanderen (G.0356.12).

References

1. Gallagher, A.C., Batra, D., Parikh, D.: Inference for order reduction in Markov random fields. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2011) 1857–1864
2. Sontag, D., Meltzer, T., Globerson, A., Jaakkola, T., Weiss, Y.: Tightening LP relaxations for MAP using message passing. In: Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence, AUAI Press (2008) 503–510
3. Domingos, P., Kok, S., Lowd, D., Poon, H., Richardson, M., Singla, P.: Chapter 'Markov Logic'. In: Probabilistic Inductive Logic Programming - Theory and Applications. Lecture Notes in Computer Science. Springer (2008)
4. Mladenov, M., Ahmadi, B., Kersting, K.: Lifted linear programming. In: Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (2012) Volume 22 of JMLR: W&CP 22.
5. Niepert, M.: Markov chains on orbits of permutation groups. In: Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence. (2012)