Reduce and Re-Lift: Bootstrapped Lifted Likelihood Maximization for MAP

Fabian Hadiji and **Kristian Kersting**
IGG, University of Bonn and KD, Fraunhofer IAIS
{firstname.lastname}@iais.fraunhofer.de

Abstract

By handling whole sets of indistinguishable objects together, lifted belief propagation approaches have rendered large, previously intractable, probabilistic inference problems quickly solvable. In this paper, we show that Kumar and Zilberstein's likelihood maximization (LM) approach to MAP inference is liftable, too, and actually provides additional structure for optimization. Specifically, it has been recognized that some pseudo marginals may converge quickly, turning intuitively into pseudo evidence. This additional evidence typically changes the structure of the lifted network: it may expand or reduce it. The current lifted network, however, can be viewed as an upper bound on the size of the lifted network required to finish likelihood maximization. Consequently, we re-lift the network only if the pseudo evidence yields a reduced network, which can efficiently be computed on the current lifted network. Our experimental results on Ising models, image segmentation and relational entity resolution demonstrate that this bootstrapped LM via "reduce and re-lift" finds MAP assignments comparable to those found by the original LM approach, but in a fraction of the time.

Introduction

Probabilistic graphical models are a powerful modeling framework for various kinds of problems. One of the most important inference tasks is finding the *maximum a posteriori* (MAP) assignment, which returns the most likely assignment to all variables in the problem. It has applications in various domains such as image processing (Yanover et al. 2006), entity resolution (Singla and Domingos 2006), and dependency parsing (McDonald et al. 2005), among others. For real-world problems, however, exact MAP inference is often intractable and one has to resort to approximate inference approaches such as message passing. Starting from Belief Propagation (BP) with max-product updates (Pearl 1991), over Linear Programming (LP) based relaxations such as MPLP (Globerson and Jaakkola 2007) and algorithms such as Tree-Reweighted BP for max-product (TRWBP) (Wainwright, Jaakkola, and Willsky 2005) or dual decomposition relaxation (Sontag, Globerson, and Jaakkola 2011) in general. In this paper, we consider a recently proposed likelihood maximization (LM) approach (Kumar and Zilberstein 2010) which was empirically shown to have a convergence rate often significantly higher than MPLP by increasing a lower bound on the MAP value. Intuitively, LM approximates the original distribution by a simpler one and uses Expectation Maximization (EM) to optimize this objective. LM can be implemented using message passing and was shown to find high quality solutions quickly. Specifically, as our first contribution, we show that LM is liftable.

Lifting inference, see e.g. (Kersting 2012) for a recent overview, essentially follows an upgrading paradigm. That is one takes a well-known propositional inference approach and adapts it to run on a more compact representation that exploits symmetries in the graphical structure of the original model for a given query at hand. Indeed, this lifting can be done "top-down" — starting at the most "lifted" level and shatter the lifted model against itself and the evidence until a modified inference approach can run safely on the model (Poole 2003; de Salvo Braz, Amir, and Roth 2005; Singla and Domingos 2008) — or in bottom-up fashion — starting at the propositional level and lift that model to obtain a representation as compact as possible (Kersting, Ahmadi, and Natarajan 2009; Niepert 2012). While bottom-up approaches need to ground every relational model first, top-down algorithms struggle with problems naturally presented in propositional form.

Indeed, showing that LM is liftable (in a bottom-up fashion) already significantly extends the family of known lifted MAP approaches, see (Apsel and Brafman 2012; Bui, Huynh, and Riedel 2012) and references in there. Our main contribution, however, is to show how to employ additional structure for optimization provided by LM to speed up lifted inference further: *pseudo evidence*. For LM, it has been recognized that pseudo marginals may converge quickly when forcing updates to be as greedy as possible (Toussaint, Charlin, and Poupart 2008). We interpret this greedy update rule as ultimately inducing pseudo evidence: *if a pseudo marginal is close to one state, clamp the corresponding variable accordingly*. Although, as we will show empirically, this can already considerably speed up inference in the propositional case, since the model could potentially be simplified over inference iterations, our naive lifted LM approach cannot make use of this new evidence to speedup lifted inference even further in later iterations. Consequently, we propose an efficient lifted LM version for updating the structure of the lifted network with pseudo evidence over the iterations.

To do so, one is tempted to just lift the network again when new pseudo evidence is "observed" using e.g. online lifted inference approaches and related approaches (Ahmadi, Kersting, and Hadiji 2010; Nath and Domingos 2010; Bui, Huynh, and de Salvo Braz 2012). However, we can do significantly better. Existing online approaches simply compute a more shattered model, should the new evidence require to do so. But note that the current lifted model is always valid for the remaining iterations. Therefore, we can simply lift the current lifted model and obtain monotonically decreasing models. So, in contrast to existing lifted inference approaches, we add evidence over the iterations and use this additional evidence for lifting. Indeed, this bootstrapping LM using pseudo evidence is akin to lifted RCR "relax, compensate and then recover" (den Broeck, Choi, and Darwiche 2012) because messages to clamped variables do not have to be calculated anymore, hence these edges could be deleted. However, whereas RCR first relaxes and then compensates, we just "reduce and rel-lift" over the iterations.

To summarize, this paper makes the following contributions: (**C1**) We show that LM can be lifted using a color passing algorithm. On the way, we also clarify for the first time that every iteration of color passing takes time linear in the number of edges. (**C2**) For MAP, we exploit pseudo evidence to save message computations without decreasing the quality of the solution drastically. (**C3**) By employing bootstrapped[1] re-lifting we compress the problem additionally in later iterations. (**C4**) We show how to re-lift solely on the lifted level without resorting to the ground model. Our experimental results on MLNs, Ising models, image segmentation and entity resolution show that bootstrapped lifted LM can yield considerable efficiency gains compared to standard LM and naive lifted LM.

We proceed as follows. We start off by reviewing LM. We then develop its naive lifted variant. Afterwards, we show how to obtain pseudo evidence and how it can be used to efficiently re-lift the lifted model. Before concluding, we will present our experimental evaluation.

Likelihood Maximization for MAP Estimation

In the following, we use upper case letters (X) to denote variables and lower case letters (x) for instantiations. Variable sets are depicted by $\mathbf{X}$ and their instantiations by $\mathbf{x}$. A Markov Random Field (MRF) over $\mathbf{X}$ is an undirected graphical model with the joint probability $P(\mathbf{X} = \mathbf{x})$ denoted as $p(\mathbf{x}) = \frac{1}{Z}\prod_k f_k(\mathbf{x_k})$ where the potentials f_k are non-negative functions over a subset $\mathbf{x_k}$ of the variables and Z is a normalization constant. An MRF can be described by an undirected graph $G = (V, E)$, where V is a set of nodes with a node for every variable in $\mathbf{X}$ and E is a set of edges with an edge for every two nodes if their corresponding variables appear together in a potential. Formally, the task of MAP inference is defined as $\arg\max_{\mathbf{x}} p(\mathbf{x})$. Every MRF can be equivalently described in terms of a factor graph. A factor graph is a bipartite graph that has a variable node for each X_i, a factor node for each f_k, and an edge connecting variable node i to factor node k if and only if X_i is an argument of f_k. In our figures, we denote variables as circles and factor nodes as squares.

Inspired by the equivalence between the MAP problem and solving a linear program, Kumar and Zilberstein place a constraint on the probability distribution in the definition of the marginal polytope. For details on the marginal polytope we refer to (Wainwright and Jordan 2008). More precisely, they look at distributions for which $p(\mathbf{x}) = \prod_{i=1}^{n} p_i(x_i)$ holds and maximize over the resulting set of mean parameters. This set of parameters defines an inner bound of the marginal polytope. Kumar and Zilberstein show that this maximization problem can be reformulated as likelihood maximization in a finite-mixture of simple Bayes nets. The hidden variables in the mixture are the X_i in the original MRF. The potentials f_k are incorporated via "binary reward variables" $\hat{f}_k$ and the conditional probability distribution of $\hat{f}_k$ is proportional to f_k, i.e., for every potential f_k in the MRF, there exists a Bayes net with a reward variable and its parents being x_k. The authors introduce an EM approach that monotonically increases the lower bound of the MAP. We will now give the EM message equations in an adapted form for arbitrary factor graphs[2]. A message sent from a factor f to its neighboring variables is defined as: $\mu_{f\to X}(x) = \sum_{\neg\{x\}} f(\mathbf{x}) \prod_{i\in \mathrm{nb}(f)\setminus X} p_i(x_i)$, where $\neg\{x\}$ denotes all possible instantiations of the variables in the domain of f with variable X fixed to x and $\mathrm{nb}(f)$ denotes the variables connected to factor f. The M-Step updates the belief of the current marginal probability in every iteration as follows: $p_i^*(x_i) = p_i(x_i) C_i^{-1} \sum_{f\in \mathrm{nb}(X)} \mu_{f\to X}(x_i)$, where C_i is a normalization constant for variable X_i. This process of sending messages is iterated until convergence. For what follows, it is important to note that a modification of this M-Step is common in order to speed up convergence: the so called *soft greedy M-Step*, originally proposed in (Toussaint, Charlin, and Poupart 2008), is used to weight the current expectation stronger.

Lifted LM (LLM)

We now derive the formulas for the lifted variant of LM. As shown in (Kersting, Ahmadi, and Natarajan 2009), lifting can be viewed as a Color Passing (CP) procedure that finds symmetries in a given factor graph by sending color messages. Intuitively, CP assigns nodes the same color that have the same distribution under a message passing algorithm, such as BP, following a parallel schedule. For the lifted LM this means that one identifies variables that have the same optimizing distribution in the EM algorithm. Due to the space restrictions, we refer the reader to (Kersting, Ahmadi, and Natarajan 2009) for more details. If we want to run the LM algorithm on the lifted model, we have to adapt the messages sent from factors to variables and the M-step,

[1] Here, we refer to bootstrapping as the capability of self-improvement during inference stage.

[2] Kumar and Zilberstein derived them for pairwise MRFs only.

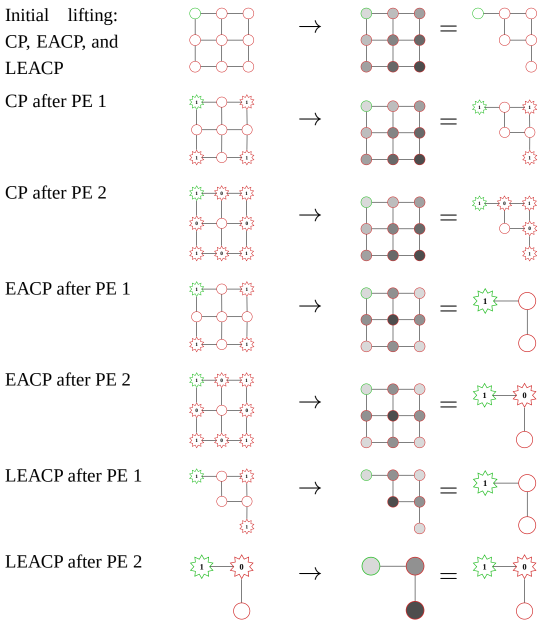


Figure 1: Re-lifting a network with PE which is *not* optimally lifted via CP if PE is present. (Best viewed in color)

to reflect merged groups of variables and factors, i.e. nodes with the same color after running CP. This is achieved by adding counts to the formulas. We can save messages sent from factors to nodes by observing that factors connected to k variables of the same color, only need to calculate a single message instead of k because all k messages are identical. With a similar argument, we multiply the incoming messages by k in the M-step if a variable is connected to k factors of the same color. Hence, the lifted message equation of the M-Step is:

$$p_i^*(\mathfrak{x}_i) = p_i(\mathfrak{x}_i) C_i^{-1} \sum\nolimits_{\mathfrak{f}\in \mathrm{nb}(\mathfrak{X})} \mu_{\mathfrak{f}\to\mathfrak{X}}(\mathfrak{x}_i) \cdot \mathrm{c}_{f,X}\,, \tag{1}$$

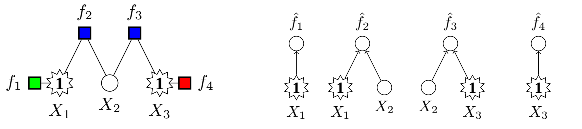
where $\mathfrak{x}_i$ are variables and $\mathfrak{f}$ are factors in the lifted network. As opposed to the ground case, the lifted equation sums over the lifted network ($\mathfrak{f} \in \mathrm{nb}(\mathfrak{X})$) and introduces the count $\mathrm{c}_{f,X}$. The count is the number of times, ground variable X is connected to factors with the color of f in the ground network. This M-Step has only to be done once for every distinct variable color. Lifted LM can be thought of sorting the Bayes nets into buckets. Initially, every Bayes net is assigned to a bucket according to the conditional probability table of $\hat{f}_k$. Then variables compute their color signatures as before. The Bayes nets are now put into different buckets depending on the colors of their child nodes.

Bootstrapped LM (BLM)

While lifting is an exact method to speed up inference, we will now introduce an additional, approximate adaption to the LM approach that reduces running times even further. When LM iteratively computes p_i^*, the uncertainty about the MAP state decreases over the iterations. Once the probability for one state is above a threshold π, one can assume that it will not change its state in the final MAP assignment anymore. Since LM essentially implements a gradient-based solver for the underlying mathematical MAP program (Kumar and Zilberstein 2010), being close to one state makes it very unlikely ever turning to a different one. In such cases, we fix the distribution in such a way that all states have zero probability except for the most likely state which is set to one. More formally, if $x_i^* = \arg\max_{x_i} p_i(x_i)$ and $p_i(x_i^*) > \pi$, we set $p_i(x_i) = 1$ if $x_i = x_i^*$ and $p_i(x_i) = 0$ if $x_i \neq x_i^*$. We call such states *Pseudo Evidence* (PE) because they do not belong to knowledge that is available beforehand but instead becomes available during the inference. More importantly, PE has major implications on future message updates. It simplifies the MRF because it cancels out states from the potentials. It allows skipping messages to these variables because for clamped variables the M-step is obsolete. Recall that the idea of PE is inspired by the soft greedy rule for the M-Step in LM and also reduces the number of iterations required to converge. On top of that, we can now combine PE and lifting. We can re-lift the network during inference based on this new evidence. Thereby we obtain a more compact lifted network. Why can this be the case? Intuitively, PE can introduce additional independencies in the graphical model which can be exploited via lifting. However, fully exploiting PE for lifting requires an adapted form of CP. Before we explain this bootstrapped lifted LM, we want to give some intuition into the approximate nature of the maximization when PE is present. Essentially there are two errors that can be introduced by PE. (**A1**) In rare cases a variable will change its state again, even though its belief was already above the threshold π. Therefore, we also propose a variant of BLM where we do not directly clamp a variable on observing a belief above π for the first time but instead we only mark it for clamping. We clamp the variable if we see that its state remains the same for the following d iterations and proceed as before. To estimate the effect of this lag, we simply count the number of state changes during the iterations for different d. (**A2**) Clamping a variable can be seen as introducing an error to the messages. But as shown in (Ihler, III, and Willsky 2005) for BP, the influence of the error on other variables in the network decays with the distance to the clamped node. Additionally, this influence does not have to be negative. In fact, it can also lead to faster converge to the correct state. Determining the exact influence of PE is difficult but we estimate it by comparing the final MAP solution obtained by BLM to the result of standard LM. In the experimental section we will empirically show that both issues do not have critical influence on the quality. Instead, BLM results are of high quality.

Bootstrapped Lifted LM (BLLM)

We now combine PE and lifting. A naive lifting approach takes the ground network, clamps it according to the PE, uses CP or any online variant to lift it, and continues message passing on this new lifted network. This approach is depicted for CP in Fig. 1 (rows 1-3). Although the approach sounds promising, the example already illustrates the downside of this naive approach. The model used in the example is a binary, pairwise MRF with identical pairwise potentials. The line color of nodes denotes different unary potentials f_i.



(a) (b)

Figure 2: Factor graph and corresponding Bayes nets. X_1 and X_3 are clamped. f_1 (green) and f_4 (red) are different while f_2 and f_3 are identical (blue). (Best viewed in color)

In our example, all unary potentials are identical (red) except for the one associated with the variable in the upper left corner (green). The first row in Fig. 1 shows the ground network without any PE. After running CP, we get the graph shown on the RHS of the arrow with nodes colored according to CP. This graph can be simplified to the lifted network next to the equality sign. Now assume that at some point we obtain identical PE for all four corners (depicted as stars instead of circles for the variable nodes). Running CP on the ground network with the new evidence in mind, we obtain the very same lifting as in the previous iteration, namely, a network containing six variables (Fig. 1, second row). This behavior of CP is sub-optimal. Since the upper left corner is set, its unary factor does not have any influence on any message being sent in future iterations. This also holds for the other three corners. This means that their unary factors can actually be neglected and all four corner nodes should receive the same color. In the third row of Fig. 1 we assume that all remaining unclamped variables are now being fixed except for the center variable. Again, CP is not able to exploit additional symmetries. We will now show how to modify CP to achieve a higher compression due to "edge-deletion" induced by PE.

EACP (Evidence Aware Color Passing) is similar to the original CP but capable of returning a lifted network that respects PE. We begin by illustrating the underlying idea by an example. Assume the factor graph in Fig. 2a has identical and symmetric pairwise potentials f_2 and f_3. The unary factors of X_1 and X_3 are different. We also assume that X_1 and X_3 have just been set to 1 (again denoted as stars). Fig. 2b shows the Bayes nets associated with the factor graph. The PE on X_1 and X_3 simplifies the messages $\mu_{f_2 \rightarrow X_2}$ and $\mu_{f_3 \rightarrow X_2}$ in such a way that they become identical. In a network without PE, f_2 and f_3 send different messages to X_2 because these messages are dependent on the current beliefs of X_1 and X_3 which are different due to their unary factors. Since the messages are now identical, we can put f_2 and f_3 into the same cluster. The key insight is that standard CP assigns too many colors, i.e., factors that actually send identical messages are colored differently. CP initializes X_1 and X_3 identically, but f_1 and and f_4 have different colors which are propagated to X_1 and X_3 in the initial color exchange. To overcome this problem, EACP differs from CP as follows. Factors only create new color signatures if they are connected to more than one unclamped variable. Otherwise they keep their current color. Nevertheless, ECAP initially propagates evidence of variables to neighboring factors, so it can distinguish identical factors connected to variables with different evidence. In turn, it can distinguish f_2 and f_3 in Fig. 2a if X_1 and X_3 had been set to different states. Secondly, PE variables never receive new colors. We now show that EACP indeed returns a correct lifting.

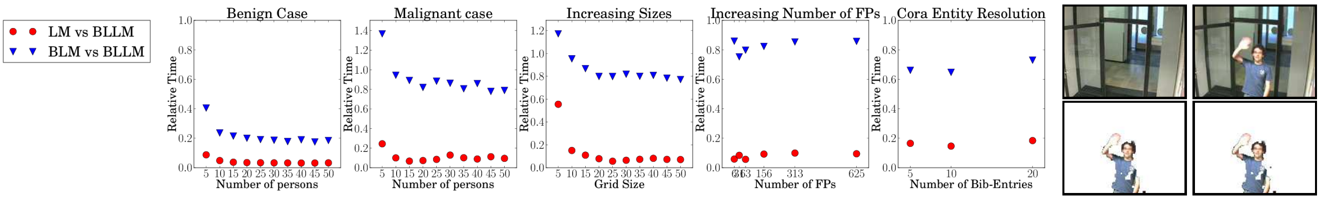
Algorithm 1: BLLM

```
cfg ← compress(fg);  /* $fg$ is input graph */
while not converged do
    calc_deltas();              /* $\mu_{f \rightarrow X}(x)$ */
    calc_marginals();           /* Eq. (1) */
    clamp_variables();
    if new pseudo evidence then
        leacp();
        average_marginals();
return calc_map();      /* MAP assignment */
```

Theorem 1. *EACP is sound, i.e., it returns a valid lifting of the ground network.*

Proof sketch. All PE variables that have the same range and are set to the same state keep their colors because their distribution is fixed. They all have the same influence and do not have to be distinguished. Nodes that have not been set behave as in standard color passing. WOLOG, let us assume pairwise factors. If one variable of a factor is clamped, messages to the other variable become independent of any of the marginal distributions; they solely depend on the potential of the factor. Hence, two factors with identical potentials receive the same colors initially and do not have to adapt their colors; the messages they send are identical which satisfies the requirement for the same color. By propagating the evidence of the variables before the actual color passing, we ensure that factors can be distinguished that are connected to PE variables in different states. □

The main idea is now to replace CP with EACP. Rows four and five in Fig. 1 exemplify how EACP obtains better compression than CP after PE is observed. However, we can still do considerably better. EACP operates on the ground network. As we will show now, it can also operate on the current lifted network. Lifted EACP (**LEACP**) essentially operates like EACP but it requires some modifications of the color signatures. We have to distinguish variables that are connected to a different number of factors of the same color. This is exactly the information contained in the counts $\mathrm{c}_{f,X}$. LEACP now simulates the ground color signature for a variable X by appending the color c of a factor f $\mathrm{c}_{f,X}$ times. This idea is only correct for lifted networks that have non-fractional counts. This makes it difficult to employ other approximate lifting approaches algorithms such as informed lifting (Kersting et al. 2010) and early stopping (Singla, Nath, and Domingos 2010). To see that LEACP returns a valid lifting, first note that setting PE will never require a cluster in the lifted network to be split. This is simply because all variables in a cluster have the same expectation of their marginal distribution so that we will always set an entire cluster to evidence. In turn, the variables in that cluster will remain to be indistinguishable in all later iterations; we will never have to split them. This essentially proves the soundness of LEACP.



(a) (b)

Figure 3: (a) Relative running time of BLLM compared to LM and BLM. (b) Input and output for the image segmentation task. In summary, BLLM outperforms all other variants in terms of running time while achieving accurate results.

Theorem 2. *LEACP is sound, i.e., it returns a valid re-lifting of the current lifting containing additional evidence.*

Putting all pieces together, we obtain **BLLM** as summarized in Alg. 1. BLLM exploits all enhancements we have introduced above. It uses CP to obtain an initial lifting (line 1). It then runs on the lifted level until PE is encountered (lines 3-5). In that case, it uses LEACP to re-lift the model on the lifted level (line 7). To complete the discussion of BLLM, we have to explain one final issue. Indeed, LEACP returns a valid lifting. However, how do we initialize the max beliefs in the re-lifted network? Newly joint nodes may have different estimated max beliefs so far. We propose to to simply take the average of the marginals of the involved variables to make use of the calculations of previous iterations (line 8 in Alg. 1). Intuitively, we do not initialize the variables again since we already know their tendency and they should get same max beliefs in the end. Although this is not exact, it does not sacrifice the estimation quality and justifies the re-lifting. This is also validated by our experimental results. It is also likely to reduce the variance of the estimates by this averaging. BLLM saves messages compared to the ground inference by avoiding the calculation of indistinguishable messages. Additionally, LEACP is more efficient than EACP because it solely operates on the lifted network to obtain the lifting for the following iterations. That is we never have to return to the ground again after the initial lifting. This is also illustrated in rows six and seven in Fig. 1 Finally, we want to comment on the time complexity of ((L)EA)CP. The idea of bootstrapped lifting only works if the repeated call to lifting is efficient. One can see that CP is a form of the 1-dimensional *Weisfeiler-Lehman* algorithm. It follows from the arguments in (Shervashidze et al. 2011) that the complexity of one iteration of CP is linear in the number of edges. Since (L)EACP involves only linear-time modifications of CP, they also run in linear time.

Experimental Evaluation

Here, we investigate the question whether our bootstrapped and lifted LM variants can be considerably faster than the baselines without decreasing performance. We answer the following main questions: (**Q1**) Does LM benefit from PE? (**Q2**) Can PE combined with lifting speed up LM even further? (**Q3**) How does the quality of B(L)LM's results compare to LM? To do so, we implemented the following set of algorithms: **(LM)** the propositional approach by Kumar and Zilberstein, **(LLM)** the naively lifted LM, **(BLM)** Bootstrapped LM using PE but no lifting, and **(BLLM)** the bootstrapped lifted LM as shown in Alg. 1. LLM, BLM, and BLLM are our contributions and will be compared to LM. Since clamping a single variable is unlikely to achieve significant gains in lifting, we re-lift after we have obtained PE for a batch of $b\%$ of the original number of variables. In our experiments we set b to 10. We stop the algorithm after $2,000$ iterations if the algorithm does not converge earlier. The other parameters are set to $\pi = 0.9$ and $d = 0$.

Smokers-MLN: MLNs (Richardson and Domingos 2006) are probabilistic relational models that are defined by weighted first-order formulas. Intuitively, the Smokers-MLN defines a social network containing friendships between persons and their smoking habits. Additionally, it contains the implication that smoking leads to cancer. Given a partial observation of the predicates, MAP inference is used to find the assignment to the remaining variables. This model is frequently used to show benefits of lifting algorithms. In our first experiment, we generate MLNs of varying domain sizes (5 up to 50), i.e. for an increasing number of persons. Resulting in factor graphs with up to 2,600 variables, 5,510 factors, and 10,150 edges. We call this setting the *Benign* Smokers-MLN because it does not contain any evidence and hence is well suited for lifting. The first plot in Fig. 3a shows that BLLM only requires a fraction of the time compared to LM (red circles). Additionally, the blue triangles highlight that lifting reduces the running required by BLM significantly. For a clearer picture, we have omitted the running times of LLM. But for this problem, LLM outperforms BLM while BLLM is the fastest method among all. We now add 25% random evidence to the MLNs. We call this setting the *Malignant* Smokers-MLN because the random evidence prevents CP from lifting the network initially. The results are depicted in the second plot in Fig. 3a. Again, BLLM only requires a fraction of LM's running time. But compared to the benign case, lifting BLM does not help as much as before. Nevertheless, lifting is still beneficial for the larger problems. For these problems, LLM basically reduces to the ground LM case as the evidence destroys initial symmetries. As shown in Tab. 1, all solutions obtained by BLLM are qualitatively as good as the LM's results.

Ising Models: Ising grids are pairwise MRFs with $x_i \in \{-1, +1\}$. The unary potentials are $f(x_i) = \theta_i x_i$ where θ_i is called the field parameter and is drawn from $[-1, 1]$. Pairwise potentials are defined as $f(x_i, x_j) = \theta_{i,j} x_i x_j$ with the

	Smokers-MLNs Benign Case									
	5	10	15	20	25	30	35	40	45	50
(L)LM	103.5	362.0	775.5	1344.0	2067.5	2946.0	3979.5	5168.0	6511.5	8010.0
B(L)LM	**103.5**	**362.0**	**775.5**	**1344.0**	**2067.5**	**2946.0**	**3979.5**	**5168.0**	**6511.5**	**8010.0**
	Smokers-MLNs Malignant Case									
	5	10	15	20	25	30	35	40	45	50
(L)LM	85.2	333.0	687.3	1234.4	1854.9	2682.3	3655.6	4744.0	5895.6	7251.8
B(L)LM	**85.2**	**333.0**	**687.3**	**1234.4**	**1854.9**	**2682.3**	**3655.6**	**4744.0**	**5895.6**	**7251.8**
	Ising-Grids of Varying Sizes									
	5	10	15	20	25	30	35	40	45	50
(L)LM	73.1	309.1	499.1	892.0	1991.6	2447.8	3095.0	3227.0	5391.7	5480.4
B(L)LM	**72.8**	**294.4**	**481.4**	**859.4**	**1913.0**	**2352.4**	**2997.2**	**3118.9**	**5201.2**	**5248.8**
	Ising-Grids of Varying Field Parameters						**Cora Entity Resolution**			
	6	31	63	156	313	625	5	10	20	
(L)LM	1693.9	1808.8	1991.6	1566.9	1093.0	1748.1	0.98	-103.6	-689.2	
B(L)LM	**1663.0**	**1739.8**	**1913.0**	**1520.4**	**1061.5**	**1681.8**	**0.98**	-116.5	**-702.8**	

Table 1: The log-scores show that the B(L)LM versions of the algorithms achieve high quality results. Bold log-scores are in 95% of the standard LM score.

interaction parameter $\theta_{i,j}$. We call the grid attractive if all $\theta_{i,j} > 0$. Motivated by applications in image processing, we use attractive interaction parameters and a limited set of field parameters instead off all θ_i drawn randomly. The first problem set consists of grids of increasing size. We generated grids from 5×5 up to 50×50 with a fixed ratio of different field parameters equal to 10%. We generate 10 grids for every size and average the results over these runs. The results are shown in the third plot in Fig. 3a. BLLM is by far the fastest method and lifting of BLM achieves around 20% speed ups on the larger instances. The second experiment keeps a fixed grid size (25×25) and varies the number of field parameters. For every grid we generated a set of field parameters beforehand. The size of this set is chosen relative to the number of variables in the grid. The fourth plot in Fig. 3a again shows that BLLM is the fastest method and requires less than 10% of LM's running time. Similar to the Malignant Smokers-MLN, lifting can here speed up BLM only little. Tab. 1 contains average log-scores for a qualitative comparison and shows that all results of BLLM are close to LM's solutions. In general, these random grids are not well suited for exact lifted inference as there is no structural symmetry. However, BLLM is still able to perform better than BLM. This clearly shows the benefits of exploiting PE and shows that lifting can become beneficial even in settings that are adversarial towards lifting in general.

Image Segmentation: To show that the BLLM algorithm actually results in high quality solutions on real world problems, we ran our algorithm on an instance of the image segmentation task. We use libDAI (Mooij 2010) to generate a factor graph for this task and we also use their example images. The inputs are the upper two images depicted in Fig. 3b. The script calculates the difference between both images to define the unary factors. The potentials of the pairwise factors are defined as in attractive Ising models, i.e. neighboring pixels are more likely to belong to the same segment. The two images in the lower row in Fig. 3b show the results after running inference (LM on the left, BLLM on the right). The images show that the result obtained by BLLM is almost as good as the one obtained by LM. One has to look very carefully to see differences. In fact, the score achieved by BLLM is only slightly lower compared to LM.

Cora: We conduct our final experiments on the Cora entity resolution dataset. We obtained the datasets from the repository at http://alchemy.cs.washington.edu and used Alchemy for weight learning. We did not evaluate on the entire dataset since subsets are already sufficient to highlight our results. Instead we randomly sampled k bib-entries and added all predicates describing their properties to the evidence. We ran experiments for $k = 5, 10,$ and 20. The largest problem instance contains 1,110 variables, 32,965 factors, and 71,209 edges. The results in Fig. 3a, 5th plot, show that BLLM clearly outperforms LM in terms of runtime, while the scores are very similar (see Tab. 1). Here, lifting achieves good performance increases over BLM again. This also holds for the naive LLM which is again omitted in the plot for clearness. We also analyzed the behavior of BLM on the Cora problem with respect to the nature of the approximation as motivated in A1 and A2. We have observed that state changes (as described in A1) during the iterations almost never occur. One has to set the lag-parameter $d > 50$ to observe any state changes at all. Similarly, as one can already guess from the log-scores, the MAP solutions are very similar and the differences seem to result from variables changing their states only but not due to wrong influence of PE.

The experimental results clearly show that LM boosted by lifting and PE can indeed considerably reduces running times without sacrificing accuracy. Moreover, BLLM scales well and is most beneficial on relational models but can also improve state-of-the-art on propositional models that so far were challenging for lifted inference approaches. To summarize, questions Q1–Q3 were all answered in favor of BLLM.

Conclusions

We introduced the idea of pseudo evidence (PE) for MAP inference and showed that it can considerably speed up MAP inference and provides novel structure for optimization. Specifically, we bootstrapped lifting using PE and empirically demonstrated that this “reduce and re-lift” can decrease running time even further and achieve lifting in situations where standard lifting would fail at all.

Bootstrapping inference using PE is not restricted to MAP inference. E.g., we investigated bootstrapped Label Propagation (LP) (Bengio, Delalleau, and Le Roux 2006) on a binary classification task as done in (Garnett et al. 2012), that is a venue was to be predicted for papers in the CiteSeer^{x} citation network. Bootstrapped LP, implementing a kind of adaptive push-back, converged to solutions essentially identical to standard LP, but in only about half the number of iterations. Exploring PE for other inference and learning approaches is an attractive avenue for future work. Other avenues tailored towards MAP inference are fixing variables for which the Markov blanket is completely set, exploiting symmetries within factors, generalizing BLLM to MRFs with non-binary variables, and approximate “reduce” steps a la edge-deletion. One should also connect PE to the use of extreme probabilities used for lifting (Choi and Amir 2012).

Acknowledgments: The authors thank the anonymous reviewers for their valuable comments. This work was partly funded by the Fraunhofer ATTRACT fellowship “STREAM”, by the DFG, KE 1686/2-1, and by the EU, FP7-248258-First-MM.

References

Ahmadi, B.; Kersting, K.; and Hadiji, F. 2010. Lifted belief propagation: Pairwise marginals and beyond. In P. Myllymaeki, T. Roos, T. J., ed., *Proceedings of the 5th European Workshop on Probabilistic Graphical Models (PGM–10)*.

Apsel, U., and Brafman, R. I. 2012. Exploiting Uniform Assignments in First-Order MPE. In *UAI*, 74–83.

Bengio, Y.; Delalleau, O.; and Le Roux, N. 2006. Label propagation and quadratic criterion. In Chapelle, O.; Schölkopf, B.; and Zien, A., eds., *Semi-Supervised Learning*. MIT Press. 193–216.

Bui, H. B.; Huynh, T. N.; and de Salvo Braz, R. 2012. Exact lifted inference with distinct soft evidence on every object. In *AAAI*.

Bui, H. H.; Huynh, T. N.; and Riedel, S. 2012. Automorphism Groups of Graphical Models and Lifted Variational Inference. In *Statistical Relational AI*.

Choi, J., and Amir, E. 2012. Lifted Relational Variational Inference. In *UAI*, 196–206.

de Salvo Braz, R.; Amir, E.; and Roth, D. 2005. Lifted first-order probabilistic inference. In *IJCAI*, 1319–1325.

den Broeck, G. V.; Choi, A.; and Darwiche, A. 2012. Lifted relax, compensate and then recover: From approximate to exact lifted probabilistic inference. In *UAI*, 131–141.

Garnett, R.; Krishnamurthy, Y.; Xiong, X.; Schneider, J.; and Mann, R. 2012. Bayesian optimal active search and surveying. In Langford, J., and Pineau, J., eds., *ICML*, 1239–1246. New York, NY, USA: Omnipress.

Globerson, A., and Jaakkola, T. 2007. Fixing max-product: Convergent message passing algorithms for map lp-relaxations. In *NIPS*.

Ihler, A. T.; III, J. W. F.; and Willsky, A. S. 2005. Loopy belief propagation: Convergence and effects of message errors. *Journal of Machine Learning Research* 6:905–936.

Kersting, K.; Ahmadi, B.; and Natarajan, S. 2009. Counting belief propagation. In Bilmes, J., and Ng, A. Y., eds., *UAI*, 277–284.

Kersting, K.; Massaoudi, Y. E.; Ahmadi, B.; and Hadiji, F. 2010. Informed lifting for message–passing. In M. Fox, D. P., ed., *AAAI*. Atlanta, USA: AAAI Press.

Kersting, K. 2012. Lifted probabilistic inference. In *ECAI*. Montpellier, France: IOS Press. (Invited Talk at the Frontiers of AI Track).

Kumar, A., and Zilberstein, S. 2010. Map estimation for graphical models by likelihood maximization. In Lafferty, J.; Williams, C. K. I.; Shawe-Taylor, J.; Zemel, R.; and Culotta, A., eds., *NIPS*, 1180–1188.

McDonald, R.; Pereira, F.; Ribarov, K.; and Hajič, J. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, 523–530. Stroudsburg, PA, USA: Association for Computational Linguistics.

Mooij, J. M. 2010. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research* 11:2169–2173.

Nath, A., and Domingos, P. 2010. Efficient lifting for online probabilistic inference. In *AAAI*.

Niepert, M. 2012. Markov chains on orbits of permutation groups. In *UAI*, 624–633.

Pearl, J. 1991. *Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 2. edition.

Poole, D. 2003. First-order probabilistic inference. In *IJCAI*, 985–991.

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

Shervashidze, N.; Schweitzer, P.; van Leeuwen, E. J.; Mehlhorn, K.; and Borgwardt:, K. M. 2011. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research* 12:2539–2561.

Singla, P., and Domingos, P. 2006. Entity resolution with markov logic. In *ICDM*, 572–582. Washington, DC, USA: IEEE Computer Society.

Singla, P., and Domingos, P. 2008. Lifted first-order belief propagation. In *AAAI*.

Singla, P.; Nath, A.; and Domingos, P. 2010. Approximate lifted belief propagation. In *Statistical Relational AI*.

Sontag, D.; Globerson, A.; and Jaakkola, T. 2011. Introduction to dual decomposition for inference. In Sra, S.; Nowozin, S.; and Wright, S. J., eds., *Optimization for Machine Learning*. MIT Press.

Toussaint, M.; Charlin, L.; and Poupart, P. 2008. Hierarchical pomdp controller optimization by likelihood maximization. In *UAI*, 562–570. Arlington, Virginia: AUAI Press.

Wainwright, M. J., and Jordan, M. I. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning* 1(1-2):1–305.

Wainwright, M. J.; Jaakkola, T.; and Willsky, A. S. 2005. Map estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory* 51(11):3697–3717.

Yanover, C.; Meltzer, T.; Weiss, Y.; Bennett, P.; and Parrado-hernndez, E. 2006. Linear programming relaxations and belief propagation an empirical study. *Journal of Machine Learning Research* 7:1887–1907.