On the Complexity and Approximation of Binary Evidence in Lifted Inference

Guy Van den Broeck
Computer Science Department
University of California, Los Angeles
guyvdb@cs.ucla.edu

Abstract

Lifted inference algorithms exploit symmetries in probabilistic models to speed up inference. They show impressive performance when calculating unconditional probabilities in relational models, but often resort to non-lifted inference when computing conditional probabilities, because the evidence breaks many of the model's symmetries. Recent theoretical results paint a grim picture, showing that conditioning on binary relations is #P-hard, and in the worst case, no lifting can be expected. In this paper, we identify *Boolean rank* of the evidence as a key parameter in the complexity of conditioning. We contrast the hardness result by showing that conditioning on binary evidence with bounded Boolean rank is efficient. This opens up the possibility of approximating evidence by a *low-rank Boolean matrix factorization* that maintains the model's symmetries and admits efficient lifted inference.

Introduction

Statistical relational models are capable of representing both *probabilistic* dependencies and *relational* structure (Getoor and Taskar 2007; De Raedt et al. 2008). Due to their first-order expressivity, they concisely represent probability distributions over a large number of propositional random variables, causing inference in these models to quickly become intractable. Lifted inference algorithms (Poole 2003) attempt to overcome this problem by exploiting *symmetries* found in the relational structure of the model.

In the absence of evidence, *exact* lifted inference algorithms work well for large classes of statistical relational models (Jaeger and Van den Broeck 2012). They perform inference that is polynomial in the number of objects in the model (Van den Broeck 2011) and are therein exponentially faster than classical inference algorithms. When conditioning a query on a set of evidence literals, however, these lifted algorithms lose their advantage over classical ones. The intuitive reason is that evidence breaks the symmetries in the model. The technical reason is that these algorithms perform an operation called *shattering*, which ends up reducing the first-order model to a propositional one. This issue is implicitly reflected in the experiment sections of exact lifted inference papers. Most report on experiments without evidence. Examples include publications on FOVE (Poole 2003; de Salvo Braz, Amir, and Roth 2005; Milch et al. 2008) and WFOMC (Van den Broeck et al. 2011; Van den Broeck 2011). Others have found ways of efficiently dealing with evidence on only *unary* predicates. They perform experiments without any evidence on binary or higher-arity relations. There are again examples for FOVE (Taghipour et al. 2012; Bui, Huynh, and de Salvo Braz 2012), WFOMC (Van den Broeck and Davis 2012), PTP (Gogate and Domingos 2011) and CP (Jha et al. 2010).

This evidence problem has largely been ignored in the exact lifted inference literature, until recently, when Bui, Huynh, and de Salvo Braz (2012) and Van den Broeck and Davis (2012) showed that conditioning on *unary* evidence is tractable. More precisely, conditioning on unary evidence is polynomial in the amount of evidence. This type of evidence expresses attributes of objects in the world, but not relations between them. Unfortunately, Van den Broeck and Davis (2012) also showed that this tractability does not extend to evidence on *binary* relations, for which conditioning on evidence is #P-hard. Even if conditioning is hard in general, its complexity should depend on properties of the specific relation that is conditioned on. It is clear that some binary evidence is easy to condition on, even if it talks about a large number of ground atoms, for example when all atoms are true ($\forall X, Y\ \mathrm{p}(X, Y)$) or false ($\forall X, Y \neg\, \mathrm{p}(X, Y)$). As our *first main contribution*, we formalize this intuition and characterize the complexity of conditioning more precisely in terms of the *Boolean rank* of the evidence. We show that it is a measure of how much lifting is possible, and that one can efficiently condition on large amounts of evidence, provided that its Boolean rank is bounded.

Despite the limitations, useful applications of exact lifted inference were found by sidestepping the evidence problem. For example, in lifted generative learning (Van den Broeck 2013), the most challenging task is to compute partition functions without evidence. Regardless, the lack of symmetries in real applications is often cited as a reason for rejecting the idea of lifted inference entirely (informally called the "death sentence for lifted inference"). This problem has been avoided for too long, and as lifted inference gains maturity, solving it becomes paramount. As our *second main contribution*, we present a first general solution to the evidence problem. We propose to approximate evidence by an over-symmetric matrix with low Boolean rank. The need for approximating evidence is new and specific to lifted inference: in (undirected) probabilistic graphical models, more evidence typically makes inference easier. Practically, we will show that existing tools from the data mining community can be used for this low-rank Boolean matrix factorization task.

The evidence problem is less pronounced in the *approximate* lifted inference literature. These algorithms often introduce approximations that lead to symmetries in their computation, even when there are no symmetries in the model. Also for approximate methods, however, the benefits of lifting will decrease with the amount of symmetry-breaking evidence. One example is CBP (Kersting, Ahmadi, and Natarajan 2009), which reports optimal performance at 0% evidence. We expect our approximation technique to also improve the performance of those algorithms.

1 Encoding Binary Relations in Unary

Our analysis of conditioning is based on a reduction, turning evidence on a binary relation into evidence on several unary predicates. We first introduce some necessary background, and then describe the reduction.

Background

An *atom* $\mathrm{p}(t_1, \dots, t_n)$ consists of a predicate $\mathrm{p}\,/n$ of arity n followed by n arguments, which are either (lowercase) *constants* or (uppercase) *logical variables*. A *literal* is an atom a or its negation $\neg a$. A formula combines atoms with logical connectives (e.g., $\vee$, $\wedge$, $\Leftrightarrow$). A formula is *ground* if it does not contain any logical variables. An Herbrand *interpretation* or *possible world* assigns a truth value to each atom.

Statistical relational languages define a probability distribution over interpretations. Many have been proposed in recent years. Our analysis will apply to all such languages, including MLNs (Richardson and Domingos 2006), parfactors (Poole 2003) and WFOMC problems (Van den Broeck et al. 2011). For these models, we will consider the tasks of computing *conditional probabilities* $\Pr(q \,|\, e)$ and *most probable explanations* (MPE) of e.

Example 1. The following MLNs model the dependencies between web pages. A first, peer-to-peer model says that student web pages are more likely to link to other student pages:

$$w \quad \mathrm{studentpage}(X) \wedge \mathrm{linkto}(X,Y) \Rightarrow \mathrm{studentpage}(Y)$$

A second, hierarchical model says that professors are more likely to link to course pages:

$$w \quad \mathrm{profpage}(X) \wedge \mathrm{linkto}(X,Y) \Rightarrow \mathrm{coursepage}(Y)$$

Evidence e is assumed to be of the form $l_1 \wedge l_2 \wedge \cdots \wedge l_n$ where l_i are ground unary (arity 1) or binary (arity 2) literals. Without loss of generality, evidence is assumed to be full (i.e, instantiating each ground atom) for all binary relations that appear in e.[1] Instead of representing the evidence as a term, we will represent unary evidence as a Boolean vector and binary evidence as a Boolean matrix. A final (weak) assumption is that the statistical relational language being used can express universally quantified hard logical constraints.

Example 2. The evidence matrix

$$\mathbf{P} = \begin{array}{c} \mathrm{p}(X,Y) \\ X=a \\ X=b \\ X=c \\ X=d \end{array} \begin{array}{c} \begin{array}{cccc} Y=a & Y=b & Y=c & Y=d \end{array} \\ \left[\begin{array}{cccc} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array}\right] \end{array}$$

represents the evidence term

$$e = \mathrm{p}(a,a) \wedge \mathrm{p}(a,b) \wedge \neg\,\mathrm{p}(a,c) \wedge \cdots \wedge \neg\,\mathrm{p}(d,c) \wedge \mathrm{p}(d,d)$$

Vector-Product Binary Evidence

Certain binary relations can be represented by a pair of unary predicates. By adding the formula

$$\forall X, \forall Y,\ \mathrm{p}(X,Y) \Leftrightarrow \mathrm{q}(X) \wedge \mathrm{r}(Y) \tag{1}$$

to our statistical relational model and conditioning on the q and r relations, we can condition on certain types of binary p relations. Assuming that we condition on the q and r predicates, adding this formula (as hard clauses) to the model does not change the probability distribution over the atoms in the original model. It is merely an indirect way of conditioning the p relation.

If we now represent these unary relations by vectors $\mathbf{q}$ and $\mathbf{r}$, and the binary relation by the binary matrix $\mathbf{P}$, the above technique allows us to condition on any relation $\mathbf{P}$ that can be factorized in the outer vector product

$$\mathbf{P} = \mathbf{q}\,\mathbf{r}^{\intercal}$$

Example 3. Consider the following outer vector factorization of the Boolean matrix $\mathbf{P}$.

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^{\intercal}$$

In a model containing Formula 1, this factorization indicates that we can condition on the 16 binary evidence literals

$$e = \neg\,\mathrm{p}(a,a) \wedge \neg\,\mathrm{p}(a,b) \wedge \cdots \wedge \neg\,\mathrm{p}(d,c) \wedge \mathrm{p}(d,d)$$

of $\mathbf{P}$ by conditioning on the the 8 unary evidence literals

$$\begin{aligned} e = &\neg\,\mathrm{q}(a) \wedge \mathrm{q}(b) \wedge \neg\,\mathrm{q}(c) \wedge \mathrm{q}(d) \\ &\wedge \mathrm{r}(a) \wedge \neg\,\mathrm{r}(b) \wedge \neg\,\mathrm{r}(c) \wedge \mathrm{r}(d) \end{aligned}$$

represented by $\mathbf{q}$ and $\mathbf{r}$.

Matrix-Product Binary Evidence

This idea of encoding a binary relation in unary relations can be generalized to n pairs of unary relations, through adding the following formula to our model.

$$\begin{aligned} \forall X, \forall Y,\ \mathrm{p}(X,Y) \Leftrightarrow\ & (\mathrm{q}_1(X) \wedge \mathrm{r}_1(Y)) \\ & \vee (\mathrm{q}_2(X) \wedge \mathrm{r}_2(Y)) \\ & \vee \dots \\ & \vee (\mathrm{q}_n(X) \wedge \mathrm{r}_n(Y)) \end{aligned} \tag{2}$$

[1] Partial evidence on the relation p can be encoded as full evidence on predicates p_0 and p_1 by adding $\forall X,Y\ \mathrm{p}(X,Y) \Leftarrow \mathrm{p}_1(X,Y)$ and $\forall X,Y\ \neg\,\mathrm{p}(X,Y) \Leftarrow \mathrm{p}_0(X,Y)$ to the model.

By conditioning on the q_i and r_i relations, we can now condition on a much richer set of binary p relations. The relations that can be expressed this way are all the matrices that can be represented by the sum of outer products (in Boolean algebra, where $+$ is $\vee$ and $1 \vee 1 = 1$):

$$\mathbf{P} = \mathbf{q}_1\, \mathbf{r}_1^\intercal \vee \mathbf{q}_2\, \mathbf{r}_2^\intercal \vee \cdots \vee \mathbf{q}_n\, \mathbf{r}_n^\intercal,$$

or equivalently

$$\mathbf{P} = \mathbf{Q}\,\mathbf{R}^\intercal, \tag{3}$$

where the columns of $\mathbf{Q}$ and $\mathbf{R}$ are the $\mathbf{q}_i$ and $\mathbf{r}_i$ vectors respectively, and the matrix multiplication is performed in Boolean algebra, that is,

$$(\mathbf{Q}\,\mathbf{R}^\intercal)_{i,j} = \textstyle\bigvee_r \mathbf{Q}_{i,r} \wedge \mathbf{R}_{j,r}$$

Example 4. Consider the following $\mathbf{P}$, its decomposition into a sum/disjunction of outer vector products, and the corresponding Boolean matrix multiplication.

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^\intercal \vee \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}^\intercal \vee \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^\intercal$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}^\intercal$$

This factorization shows that it is possible to condition on the binary evidence literals of $\mathbf{P}$ by conditioning on the unary literals

$$\begin{aligned} e = {} & \neg\, q_1(a) \wedge \quad q_1(b) \wedge \neg\, q_1(c) \wedge \quad q_1(d) \\ & \wedge \quad r_1(a) \wedge \neg\, r_1(b) \wedge \neg\, r_1(c) \wedge \quad r_1(d) \\ & \wedge \quad q_2(a) \wedge \quad q_2(b) \wedge \neg\, q_2(c) \wedge \neg\, q_2(d) \\ & \wedge \quad r_2(a) \wedge \quad r_2(b) \wedge \neg\, r_2(c) \wedge \neg\, r_2(d) \\ & \wedge \neg\, q_3(a) \wedge \neg\, q_3(b) \wedge \quad q_3(c) \wedge \neg\, q_3(d) \\ & \wedge \neg\, r_3(a) \wedge \neg\, r_3(b) \wedge \quad r_3(c) \wedge \neg\, r_3(d) \end{aligned}$$

2 Boolean Matrix Factorization

Matrix factorization (or decomposition) is a popular linear algebra tool. Some well-known instances are singular value decomposition and non-negative matrix factorization (NMF) (Seung and Lee 2001; Berry et al. 2006), which decomposes a matrix into a product of matrices with only non-negative entries. NMF attracted much attention recently, in unsupervised learning and feature extraction, because its decompositions do not contain negative numbers and are therefore more easily interpretable. These factorizations all work with real-valued matrices. We instead consider Boolean-valued matrices, with only 0/1 entries.

Boolean Rank

Factorizing a matrix $\mathbf{P}$ in Boolean algebra as $\mathbf{Q}\,\mathbf{R}^\intercal$ is a known problem called Boolean Matrix Factorization (BMF) (Miettinen et al. 2006; Snášel, Platoš, and Krömer 2008; Miettinen et al. 2008). BMF factorizes a $(k \times l)$ matrix $\mathbf{P}$ into a $(k \times n)$ matrix $\mathbf{Q}$ and a $(l \times n)$ matrix $\mathbf{R}$, where potentially $n \ll k$ and $n \ll l$ and we always have that $n \leq \min(k, l)$.

Any Boolean matrix can be factorized this way and the smallest number n for which it is possible is called the *Boolean rank* of the matrix. Unlike (textbook) real-valued rank, computing the Boolean rank of a matrix is NP-hard and cannot be approximated unless P=NP (Miettinen et al. 2006). The Boolean and real-valued rank of a matrix are incomparable, and the Boolean rank can be exponentially smaller than the real-valued rank.

Example 5. The factorization in Example 4 is a BMF with Boolean rank 3. It is only a decomposition in Boolean algebra and not over the real numbers. Indeed, the matrix product over the reals contains an incorrect value of 2:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \times_{real} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}^\intercal = \begin{bmatrix} 1 & 1 & 0 & 0 \\ \mathbf{2} & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \neq \mathbf{P}$$

Note that $\mathbf{P}$ is of full real-valued rank (having four non-zero singular values) and that its Boolean rank is lower than its real-valued rank.

Approximate Boolean Factorization

The problem of computing Boolean ranks is a theoretical one. Because many real-world matrices will have close to full Boolean rank, applications of BMF look at approximate factorizations, where the matrix is approximated as a product of two significantly smaller matrices. The goal is to find

$$\underset{\mathbf{Q}_{k\times n}, \mathbf{R}_{l\times n}}{\arg\min}\ \mathrm{d}\left(\mathbf{P}_{k\times l}\,,\, \left(\mathbf{Q}_{k\times n}\,\mathbf{R}_{l\times n}^\intercal\right)\right)$$

for some distance function d. When $n \ll k$ and $n \ll l$, this approximation extracts interesting structure from the matrix and removes noise. For this reason, BMF has recently received considerable attention in the data mining community, as a tool for analyzing high-dimensional data. In this data mining research, the goal is to find interpretable (here meaning Boolean, not real-valued) representations of the most important concepts in a data matrix.

Unfortunately, the approximate BMF optimization problem is NP-hard as well, and inapproximable (Miettinen et al. 2008). However, several algorithms have been proposed that work well in practice. Algorithms exist that find good approximations for fixed values of n (Miettinen et al. 2008) and that select optimal values of n using the MDL principle, (Miettinen and Vreeken 2011). When $\mathbf{P}$ is sparse, good approximations can be found in polynomial time (Miettinen 2010). BMF is related to other data mining tasks, such as biclustering (Mirkin 1996) and tiling databases (Geerts, Goethals, and Mielikäinen 2004), whose algorithms could also be used for approximate BMF.

3 Complexity of Binary Evidence

The previous sections have shown how binary evidence can be represented by unary evidence and how this corresponds to a BMF. This section builds on that insight to obtain a complexity result for conditioning on binary evidence e, that is, computing $\Pr(q \mid e)$ for ground atoms q, or the most probable explanation of e. Our analysis will apply to classes of models that are domain-liftable (Van den Broeck 2011), such as the ones in Jaeger and Van den Broeck (2012), where no grounding is required and inference runs polynomial in the number of objects.

Unary Evidence

Bui, Huynh, and de Salvo Braz (2012) and Van den Broeck and Davis (2012) both present lifted algorithms for conditioning on *unary* relations. Their algorithms have a complexity that is polynomial in the number of evidence atoms in e. The technique of Van den Broeck and Davis (2012) supports hard evidence on multiple different predicates. It is not clear whether Bui, Huynh, and de Salvo Braz (2012) also supports this, but they do allow for soft evidence on a single predicate. We refer to these papers for a more technical discussion, and provide the following intuition instead.

Lifted inference algorithm exploit the fact that certain groups of objects (referred to by constants) in the world are indistinguishable and can be reasoned about as a whole. The set of constants can be grouped into equivalence classes. Each unary predicate u that we have evidence on splits a set of constants into three: for a constant c, $\mathrm{u}(c)$ is either true, false, or unobserved in the evidence. When there are n unary predicates, this induces 3^n equivalence classes of constants. Therefore, exact lifted inference algorithms are generally exponential in the number of unary predicates. However, their complexity is polynomial in the number of constants inside each equivalence class (and the number of objects in the world). For lifted algorithms, it only matters how many constants there are in each equivalence class, since they have the same properties.

Binary Evidence

It has been shown that the same tractability does not extend to binary evidence. Unless P=NP, we cannot condition on binary evidence in time polynomial in the size of the evidence term e (Van den Broeck and Davis 2012, Thm. 7). Still, it is clear that certain types of evidence are not a problem. For example, conditioning on the relation $\mathrm{p}(X, Y)$ being false for all X and Y is not exponential in the number of ground atoms being conditioned on. Regardless of the size of e, we can condition on this evidence by adding the formula $\forall X, Y, \neg\, \mathrm{p}(X, Y)$ to our model. So clearly, specific properties of the evidence do allow for tractable conditioning.

Section 1 showed that binary evidence can be represented by unary evidence, by extending the statistical relational model with pairs of unary predicates $(\mathrm{q}_i, \mathrm{r}_i)$ and with Formula 2.[2] This involves decomposing the evidence matrix $\mathbf{P}$ into a set of vector pairs $(\mathbf{q}_i, \mathbf{r}_i)$. It follows from Equation 3 and Section 2 that the number of vector pairs (and predicates) added to the model is the Boolean rank n of $\mathbf{P}$. Using the complexity results for unary conditioning, we can then condition on the specific $(\mathbf{q}_i, \mathbf{r}_i)$ vectors in time polynomial in their size. This leads us to conclude that the complexity of conditioning on a binary relation, using for instance the algorithm of Van den Broeck and Davis (2012), is not exponential in the size of the evidence, but exponential in the Boolean rank of the evidence.

Theorem 1. *The complexity of conditioning on a binary relations with bounded Boolean rank is polynomial in the size of the evidence.*

After extending a model with Formula 2 for some n, it is possible to condition on binary evidence matrices of any size in polynomial time, as long as they have Boolean rank n.

Bounded treewidth is another property of the model that permits efficient inference. Bounded Boolean rank seems to be a fundamentally different property, more related to the presence of symmetries than treewidth, which reflects sparsity. Note that the complexity of exactly computing treewidth and Boolean rank are both NP-hard (cf. Section 2).

4 Over-Symmetric Evidence Approximation

Theorem 1 opens up many new possibilities. Even for evidence with high Boolean rank, it is possible to find a *low-rank approximate BMF* of the evidence, as is commonly done for other data mining and machine learning problems. Algorithms already exist for solving this task (cf. Section 2).

Example 6. The evidence matrix from Example 4 has Boolean rank three. Dropping the third pair of vectors reduces the Boolean rank to two.

$$\begin{bmatrix}1&1&0&0\\1&1&0&1\\0&0&1&0\\1&0&0&1\end{bmatrix} \approx \begin{bmatrix}0\\1\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\0\\1\end{bmatrix}^{\intercal} \vee \begin{bmatrix}1\\1\\0\\0\end{bmatrix}\begin{bmatrix}1\\1\\0\\0\end{bmatrix}^{\intercal} \vee \xcancel{\begin{bmatrix}0\\0\\1\\0\end{bmatrix}\begin{bmatrix}0\\0\\1\\0\end{bmatrix}^{\intercal}}$$

$$= \begin{bmatrix}0&1\\1&1\\0&0\\1&0\end{bmatrix}\begin{bmatrix}1&1\\0&1\\0&0\\1&0\end{bmatrix}^{\intercal} = \begin{bmatrix}1&1&0&0\\1&1&0&1\\0&0&\mathbf{0}&0\\1&0&0&1\end{bmatrix}$$

This factorization is approximate, as it flips the evidence for atom $\mathrm{p}(c, c)$ from true to false (represented by the bold 0). By paying this price, the evidence has more symmetries, and we can condition on the binary relation by introducing only two instead of three new pairs $(\mathrm{q}_i, \mathrm{r}_i)$ of unary predicates.

Low-rank approximate BMF is an instance of a more general idea; that of *over-symmetric evidence approximation*. This means that when we want to compute $\Pr(q \mid e)$, we approximate it by computing $\Pr(q \mid e')$ instead, with evidence e' that permits more efficient inference. In this case, it is more efficient because it maintains more symmetries of the model and permits more lifting. Because all lifted inference algorithms, exact or approximate, exploit symmetries, we expect this general idea, and low-rank approximate BMF in particular, to improve the performance of any lifted inference algorithm.

Having a small amount of incorrect evidence in the approximation need not be a problem. As these literals are not covered by the first most important vector pairs, they can be

[2] Note that Formula 2 contains two logical variables. Hence, it is in the class of models for which we can prove that domain-lifted inference (polynomial in the number of objects in the world) is always possible (Van den Broeck 2011; Taghipour et al. 2013)

considered as noise in the original matrix. Hence, a low-rank approximation may actually improve the performance of, for example, a lifted collective classification algorithm. On the other hand, the approximation made in Example 6 may not be a good one if we are querying attributes of the constant c, and we might prefer to make approximations in other areas of the evidence matrix instead. There are many challenges in finding appropriate over-symmetric evidence approximations, which makes the task all the more interesting.

5 Empirical Illustration

To complement the theoretical analysis from the previous sections, we will now report on preliminary experiments that investigate the following practical questions.

Q1 How well can we approximate a real-world relational data set by a low-rank Boolean matrix?

Q2 For which Boolean ranks can we perform inference with a state-of-the-art exact lifted inference algorithm?

To answer **Q1**, we compute approximations of the linkto binary relation in the WebKB data set using the ASSO algorithm for approximate BMF (Miettinen et al. 2008). The WebKB data set consists of web pages from the computer science departments of four universities (Craven and Slattery 2001). The data has information about words that appear on pages, labels of pages and links between web pages (linkto relation). There are four folds, one for each university. The exact evidence matrix for the linkto relation ranges in size from 861 by 861 to 1240 by 1240. Its real-valued rank ranges from 384 to 503. Performing a BMF approximation adds or removes hyperlinks between web pages, so that more web pages can be grouped together that behave similarly. As discussed in the previous section, this is a good idea under the assumption that the added or removed hyperlinks are noise in the original data.

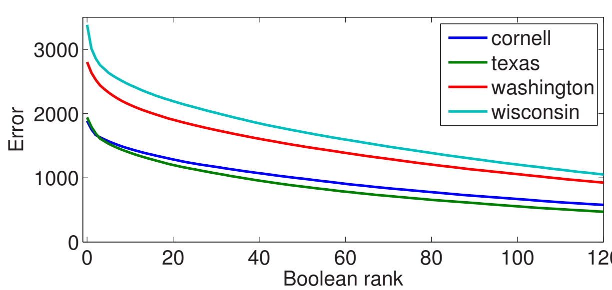

Figure 1: Approximation error (number of incorrect literals) for increasing Boolean rank of the linkto relation

Figure 1 plots the approximation error for increasing Boolean ranks. The error is measured as the number of incorrect evidence literals, or equivalently, the number of bits flipped in the matrix. The point plotted for Boolean rank 0 is the error when all literals are conditioned to be false. As these matrices are very sparse, this already yields a very low error. However, by increasing the Boolean rank only slightly, the error in the approximation reduces significantly. After a rank of around 70 to 80, the error is reduced by half, even though the matrix dimensions and real-valued rank are much higher. Note that the evidence matrix contains around a million entries, and that the approximation at rank zero already correctly labels 99.7% to 99.8% of the evidence. The highest reported ranks achieve an accuracy of 99.9% to 99.95%.

To answer **Q2**, we investigate the influence of adding Formula 2 to the "peer-to-peer" and "hierarchical" MLNs from Example 1 (also in the WebKB domain). We extend these models with Formula 2 to condition on linkto relations with increasing Boolean rank n. These models are then compiled using the WFOMC (Van den Broeck et al. 2011) algorithm into first-order NNF circuits. With these circuits, lifted inference is possible in time polynomial in the domain size, and in this case, the size of any evidence of rank n.[3]

n	0	1	2	3	4	5	6
(a)	18	58	160	1873	>2129	?	?
(b)	24	50	129	371	1098	3191	9571

Table 1: First-order NNF circuit size (number of nodes) for increasing Boolean rank n, and (a) the peer to peer and (b) hierarchical model

Table 1 shows the sizes of these circuits. As can be expected, circuit sizes seem to grow exponentially with n. Evidence appears to break more symmetries in the peer-to-peer model than in the hierarchical model, causing the circuit size to increase more quickly with Boolean rank.

6 Conclusions

We presented two main results. The first is a precise complexity characterization of conditioning on binary evidence, in terms of its Boolean rank. The second is a technique to approximate binary evidence by a low-rank Boolean matrix factorization. This is a first type of over-symmetric evidence approximation that can speed up lifted inference.

In the context of social network analysis, our decomposition approach is related to stochastic block models (Holland, Laskey, and Leinhardt 1983) and their extensions. For future work, we want to thoroughly evaluate the practical implications of the theory developed here. This includes investigating the tradeoff between approximation quality and efficiency given by the Boolean rank parameter. There are many remaining challenges in finding good evidence-approximation schemes, including ones that are query-specific (cf. de Salvo Braz et al. (2009)) or that incrementally run inference to find better approximations (cf. Kersting et al. (2010)). Furthermore, we want to investigate other subsets of binary relations for which conditioning could be efficient, in particular functional relations $\mathrm{p}(X, Y)$, where each X has at most a limited number of associated Y values.

[3]Note that having a compiled circuit does not mean that lifted inference is possible for any evidence matrix and domain size. It only means that the complexity of lifted inference is polynomial in the size of these inputs. The degree of the polynomial may be high, and inference may only be possible for small domain sizes and evidence matrices.

Acknowledgments

This work has been partially supported by ONR grant #N00014-12-1-0423, NSF grant #IIS-1118122, NSF grant #IIS-0916161, and the Research Foundation-Flanders (FWO-Vlaanderen). The author would like to thank Jilles Vreeken and Pauli Miettinen for helpful suggestions.

References

Berry, M. W.; Browne, M.; Langville, A. N.; Pauca, V. P.; and Plemmons, R. J. 2006. Algorithms and applications for approximate nonnegative matrix factorization. In *Computational Statistics and Data Analysis*, 155–173.

Bui, H.; Huynh, T.; and de Salvo Braz, R. 2012. Exact lifted inference with distinct soft evidence on every object. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*.

Craven, M., and Slattery, S. 2001. Relational learning with statistical predicate invention: Better models for hypertext. *Machine Learning Journal* 43(1/2):97–119.

De Raedt, L.; Frasconi, P.; Kersting, K.; and Muggleton, S., eds. 2008. *Probabilistic inductive logic programming: theory and applications*. Berlin, Heidelberg: Springer-Verlag.

de Salvo Braz, R.; Amir, E.; and Roth, D. 2005. Lifted first-order probabilistic inference. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1319–1325.

de Salvo Braz, R.; Natarajan, S.; Bui, H.; Shavlik, J.; and Russell, S. 2009. Anytime lifted belief propagation. *Proceedings of the 6th International Workshop on Statistical Relational Learning*.

Geerts, F.; Goethals, B.; and Mielikäinen, T. 2004. Tiling databases. In *Discovery science*, 278–289. Springer.

Getoor, L., and Taskar, B., eds. 2007. *An Introduction to Statistical Relational Learning*. MIT Press.

Gogate, V., and Domingos, P. 2011. Probabilistic theorem proving. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI)*, 256–265.

Holland, P. W.; Laskey, K. B.; and Leinhardt, S. 1983. Stochastic blockmodels: First steps. *Social networks* 5(2):109–137.

Jaeger, M., and Van den Broeck, G. 2012. Liftability of probabilistic inference: Upper and lower bounds. In *Proceedings of the 2nd International Workshop on Statistical Relational AI,*.

Jha, A.; Gogate, V.; Meliou, A.; and Suciu, D. 2010. Lifted inference seen from the other side: The tractable features. In *Proceedings of the 24th Conference on Neural Information Processing Systems (NIPS)*.

Kersting, K.; Ahmadi, B.; and Natarajan, S. 2009. Counting belief propagation. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*, 277–284.

Kersting, K.; El Massaoudi, Y.; Ahmadi, B.; and Hadiji, F. 2010. Informed lifting for message-passing. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence,*.

Miettinen, P., and Vreeken, J. 2011. Model order selection for Boolean matrix factorization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 51–59. ACM.

Miettinen, P.; Mielikäinen, T.; Gionis, A.; Das, G.; and Mannila, H. 2006. The discrete basis problem. In *Knowledge Discovery in Databases*. Springer. 335–346.

Miettinen, P.; Mielikainen, T.; Gionis, A.; Das, G.; and Mannila, H. 2008. The discrete basis problem. *IEEE Transactions on Knowledge and Data Engineering* 20(10):1348–1362.

Miettinen, P. 2010. Sparse Boolean matrix factorizations. In *IEEE 10th International Conference on Data Mining (ICDM)*, 935–940. IEEE.

Milch, B.; Zettlemoyer, L.; Kersting, K.; Haimes, M.; and Kaelbling, L. 2008. Lifted probabilistic inference with counting formulas. *Proceedings of the 23rd AAAI Conference on Artificial Intelligence* 1062–1068.

Mirkin, B. G. 1996. *Mathematical classification and clustering*, volume 11. Kluwer Academic Pub.

Poole, D. 2003. First-order probabilistic inference. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI),*, 985–991.

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

Seung, D., and Lee, L. 2001. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems* 13:556–562.

Snášel, V.; Platoš, J.; and Krömer, P. 2008. Developing genetic algorithms for Boolean matrix factorization. *Databases, Texts* 61.

Taghipour, N.; Fierens, D.; Davis, J.; and Blockeel, H. 2012. Lifted variable elimination with arbitrary constraints. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*.

Taghipour, N.; Fierens, D.; Van den Broeck, G.; Davis, J.; and Blockeel, H. 2013. Completeness results for lifted variable elimination. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics,*.

Van den Broeck, G., and Davis, J. 2012. Conditioning in first-order knowledge compilation and lifted probabilistic inference. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence,*.

Van den Broeck, G.; Taghipour, N.; Meert, W.; Davis, J.; and De Raedt, L. 2011. Lifted probabilistic inference by first-order knowledge compilation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2178–2185.

Van den Broeck, G. 2011. On the completeness of first-order knowledge compilation for lifted probabilistic inference. In *Advances in Neural Information Processing Systems 24 (NIPS),*, 1386–1394.

Van den Broeck, G. 2013. *Lifted Inference and Learning in Statistical Relational Models*. Ph.D. Dissertation, KU Leuven.