Probabilistic Reasoning with Undefined Properties in Ontologically-Based Belief Networks

Chia-Li Kuo[†] **David Buchman**[†] **Arzoo Katiyar**[‡*] **David Poole**[†]

[†] Department of Computer Science, University of British Columbia, Vancouver

[‡] Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur

[†] {kuo000, davidbuc, poole}@cs.ubc.ca [‡] arzoo@iitk.ac.in

Abstract

This paper concerns building probabilistic models with an underlying ontology that defines the classes and properties used in the model. In particular, it considers the problem of reasoning with properties that may not always be defined. Furthermore, we may even be uncertain about whether a property is defined for a given individual. One approach is to explicitly add a value "undefined" to the range of random variables, forming *extended belief networks*; however, adding an extra value to a random variable's range has a large computational overhead. In this paper, we propose an alternative, *ontologically-based belief networks*, where all properties are only used when they are defined, and we show how probabilistic reasoning can be carried out without explicitly using the value "undefined" during inference. We prove this is equivalent to reasoning with the corresponding extended belief network and empirically demonstrate that inference becomes more efficient.

1 Introduction

In many fields, we want ontologies to define the vocabulary and probabilistic models to make predictions. For example, in geology, the need to define standardized vocabularies becomes clear when one looks at detailed geological maps, which do not match at political boundaries because jurisdictions use incompatible vocabularies. There has been much recent effort to define ontologies for geology and use them to describe data (e.g., `http://onegeology.org/`). Geologists need to make decisions under uncertainty, and so need probabilistic models that use ontologies. Similarly, in the biomedical domain, huge ontologies (e.g., `http://obofoundry.org/`) are being developed and need to be integrated into decision making under uncertainty.

In an ontology, the domain of a property specifies the individuals for which the property is defined. One of the problems in the integration of ontologies and reasoning under uncertainty arises when properties have non-trivial domains and thus are not always defined. Properties applied to individuals correspond to random variables in the probabilistic model, giving rise to random variables that are not always defined. For example, the property *education* may be applicable to people but not to dogs or rocks. However, we may not know if an individual is a person, and performance on some task may depend on his/her education level if the individual is a person, such as when some damage may have been caused by a crafty person (depending on his/her education level) or by a natural phenomenon. Similarly, hardness measure (in Mohs scale) may be applicable to rocks but not to people. When modelling and learning, we do not want to think about undefined values, and we will never observe an undefined value in a dataset that obeys the ontology; for instance, we do not want to consider the education level of rocks, and no data would contain such information.

We propose a simple framework, *ontologically-based belief networks* (OBBNs), which integrates belief networks [Pearl, 1988] with ontologies. OBBNs do not explicitly use an undefined value in the construction of random variables, but by leveraging the ontology, we can compute the posterior distribution of any random variable, including the possibility of it being undefined. We define three inference methods for computing the posterior distribution of a random variable. The first method explicitly includes an extra value "undefined" in the range of the random variables that are potentially undefined. The second does not include "undefined", but involves two separate inference steps: (i) determining the probability that the query is well-defined and (ii) querying a random variable as if the well-definedness of the query were observed evidence. The third only adds "undefined" to the range of the target variable at query time.

2 Ontologies

An ontology is a formal specification of the meanings of the symbols in an information system [Smith, 2003]. Ontologies provide researchers and practitioners with standardized vocabulary in which to describe the world. An ontology defines any terminology that needs to be shared among datasets.

Modern ontologies, such as those in the Web Ontology Language (OWL) [Grau *et al.*, 2008; Motik *et al.*, 2012b], are defined in terms of classes, properties, and individuals. The semantics of OWL is defined in terms of sets [Motik *et al.*, 2012a] – a class characterizes the set of all possible in-

[*]The author contributed to this work while she was doing a summer internship at UBC.

dividuals it may contain, and a property is a binary relation, which is the set of all ordered pairs for which the property holds.

Properties have domains and ranges. The domain of a property is the class of individuals for which the property is defined. Thus, a property is only defined for individuals in its domain; for other individuals, the property is not defined. The range of a property specifies the set of possible values for the property.

It has been advocated that ontologies be written in terms of Aristotelian definitions [Aristotle, 350 BC; Berg, 1982; Poole *et al.*, 2009], where each explicitly specified class is defined in terms of a super-class, the *genus*, and restrictions on properties (e.g., that some property has a particular value), the *differentia*, that distinguish this class from other subclasses of the genus. This is not a restrictive assumption, as we can always define a property that is true of members of a class. It allows us to just model properties, with classes being defined in terms of property restrictions. So, we can model uncertainty in relationship to properties.

Thing is the single top-level class, which every other class is a subclass of, and that contains all possible individuals.

Example 1. The functional property *education* is defined for humans, and its range is $\{low, high\}$. This can be specified in OWL-2 functional syntax:

```
Declaration(ObjectProperty(:education))
FunctionalObjectProperty(:education)
ObjectPropertyDomain(:education :Human)
ObjectPropertyRange(:education
                    ObjectUnionOf(:Low :High))
```

A human is any individual in the class *Animal* for which the property *isHuman* is true:

```
Declaration(Class(:Human))
EquivalentClasses(
  :Human
  ObjectIntersectionOf(
    DataHasValue(:isHuman "true"^^xsd:boolean)
    :Animal))
Declaration(DataProperty(:isHuman))
FunctionalDataProperty(:isHuman)
DataPropertyDomain(:isHuman owl:Animal)
DataPropertyRange(:isHuman xsd:boolean)
```

Here, we stated that *isHuman* is only defined for animals. This makes the example later more interesting. An animal is an individual for which the property *isAnimal* is true and can be defined in a similar way. The domain of *isAnimal* is *Thing* and its range is a Boolean value.

In Example 1, the genus and differentia of *Animal* are *Thing* and $isAnimal = true$, respectively, whereas those of the class *Human* are *Animal* and $isHuman = true$. The property *education* has *Human* as its domain and is thus undefined when applied to an individual for which *isHuman* is not true. Likewise, *isHuman* has domain *Animal* and so is only defined for individuals of the class *Animal*.

Definition 1. An **ontology** O contains a set of logical assertions about a set $\boldsymbol{C}$ of classes, a set $\boldsymbol{Prop}$ of properties, and a set $\boldsymbol{I}$ of individuals. Each property $p \in \boldsymbol{Prop}$ has a domain and a range. In addition, O does not contain cyclic definitions of property domains in the sense that any class that defines the domain of a property cannot be defined in terms of that property.

We use $O.\boldsymbol{C}$ and $O.\boldsymbol{Prop}$ to refer to the classes and properties, respectively, of ontology O.

An **Aristotelian ontology** is an ontology where every class $C \in \boldsymbol{C}$ is either *Thing* or is defined as $C' \in \boldsymbol{C}$ conjoined with a set of property restrictions (such as $p_1 = v_1 \wedge \cdots \wedge p_k = v_k$, where $p_i \in \boldsymbol{Prop}$, $v_i \in range(p_i)$, and p_i is defined for individuals in C' for which $p_1 = v_1 \wedge \cdots \wedge p_{i-1} = v_{i-1}$). We can always reduce a class C to a conjunction of property restrictions (with *Thing* omitted), which we refer to as the **primitive form** of class C. We denote by $dom(p)$ the primitive form of the domain of p. The ontology induces a partial ordering of the properties: for every property $p \in \boldsymbol{Prop}$, any property p' appearing in $dom(p)$ precedes p, written as $p' \prec p$. Aristotelian definitions give rise to a class hierarchy and reduce class definitions to property restrictions. In Example 1, $dom(education) \equiv isAnimal = true \wedge isHuman = true$.

The acyclic restriction avoids the possibility of defining two classes in terms of each other. It ensures that to determine whether a property is defined for an individual cannot involve applying the property to that individual.

A **formula**, i.e., logical sentence written in the language (e.g., OWL) used for an ontology, is entailed by the ontology if the formula is always true given the set of logical assertions in the ontology. As a simple example, if an ontology specifies that *Student* is a subclass of *Human* and *Human* is a subclass of *Animal*, then the ontology entails that *Student* is a subclass of *Animal*, i.e., any individual in the class *Student* is also in the class *Animal*. There exist ontology reasoners for OWL, (e.g., Pellet[1] and HermiT[2]), that determine such entailments. We write $O \models \alpha$ if formula α is entailed by ontology O.

3 Integration of Ontologies with Probabilistic Models

Graphical models [Pearl, 1988] represent factorizations of joint probability distributions in terms of graphs that encode conditional (in)dependencies amongst random variables. A belief network is a directed graphical model where the factorization represents conditional probability distributions (CPDs) of the random variables. A relational probabilistic model [Getoor and Taskar, 2007; De Raedt *et al.*, 2008], also referred to as a template-based model [Koller and Friedman, 2009], extends a graphical model with the relational (or first-order) component and can be defined in terms of parametrized random variables. Given a population of individuals, such models can be grounded into standard graphical models. Our work concerns directed graphical models.

We will follow the integration of ontologies and probabilistic models of Poole *et al.* [2008; 2009], where random variables are constructed from properties and individuals. When modelling uncertainty, a functional property applied to an individual becomes a random variable, where the range of the random variable is the range of the property. A

[1] http://clarkparsia.com/pellet

[2] http://hermit-reasoner.com/

non-functional property applied to an individual becomes a Boolean random variable for each value in the range of the property. We can arrange these random variables into a DAG which specifies (conditional) dependencies.

The underlying property of a random variable is undefined for individuals not in its domain. We need a way to represent that a random variable is defined only in some contexts. As Poole *et al.* [2009] noticed, the idea of using a random variable only in a context in which it is defined is reminiscent of context-specific independence [Boutilier *et al.*, 1996; Zhang and Poole, 1999]. Here, instead of context-specific independence, a random variable is simply not defined in some contexts.

While an ontologically-based probabilistic model is inherently relational (as the ontology is about properties of individuals), the issues considered do not depend on the relational structure. For the rest of the paper, we discuss the propositional version, which can be considered as either being about a single distinguished individual or about the grounding of a relational network, with the individuals implicit.

Definition 2. For each random variable X, the **extended variant** of X is a random variable X^+ with $range(X^+) = range(X) \cup \{\perp\}$, where $X^+ = X$ when X is defined, and $X^+ = \perp$ when X is undefined.

A straightforward approach to dealing with undefined properties is to do inference using extended random variables and standard inference methods. In this paper, we explore representing and reasoning in terms of the non-extended random variables. Exact inference methods such as variable elimination [Zhang and Poole, 1994; Dechter, 1996] or recursive conditioning [Darwiche, 2001; 2009] have complexity $\mathcal{O}(|\text{variables' range}|^T)$, where T is the network treewidth. Reducing the ranges of random variables can have a dramatic impact on inference speed.

The alternatives are best given in terms of a motivating example:

Example 2. Continuing Example 1, suppose the ontology contains the property *causeDamage*, whose domain is *Thing* and range is $\{true, false\}$, which specifies whether an individual is capable of causing some particular damage.

Suppose we have the following conditional probabilities for *causeDamage*. For non-animals, there is a small probability (e.g., 0.1) that $causeDamage = true$. For animals that are not human, the probability of $causeDamage = true$ is higher (e.g., 0.3). For humans, the distribution of *causeDamage* depends on *education*. When *education* is *high*, the probability of $causeDamage = true$ is 0.9, and when *education* is *low*, the probability of $causeDamage = true$ is 0.5. A tree representation of the CPDs for *causeDamage* given its parents *isAnimal*, *isHuman*, and *education* is given in Figure 1.

Note that the conditional probabilities in Example 2 obey the constraint that a property is only used in a context where it is defined.

One way of encoding the possibility of a property being undefined is to build a model using the extended random variables. The ontology implies the probabilities:

$$P(isHuman^+ = \perp \mid isAnimal^+ = v) = 1,\ v \in \{false, \perp\}$$
$$P(education^+ = \perp \mid isHuman^+ = v) = 1,\ v \in \{false, \perp\}$$

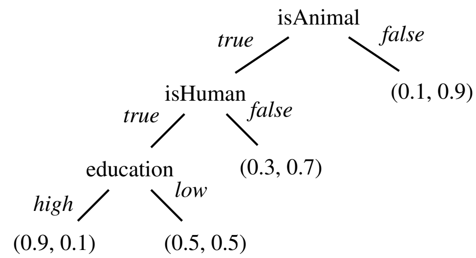

Figure 1: A tree-structured representation for $P(causeDamage \mid isAnimal, isHuman, education)$. The leaf nodes specify probability distributions over $\{true, false\}$ for *causeDamage*.

To define the distribution over all random variables, we also need the probabilities: $P(isAnimal^+ = true)$, $P(isHuman^+ = true \mid isAnimal^+ = true)$, and $P(education^+ = high \mid isHuman^+ = true)$. A belief network using the extended random variables is shown in Figure 2, where $causeDamage^+$ depends on all three other random variables. A tabular version of the CPDs for $causeDamage^+$ would define the probability distribution of $causeDamage^+$ in 27 contexts (3^3). A tree-structured version of the CPDs of $causeDamage^+$ is more complicated than that in Figure 1, because it needs to model the undefined cases that are implied by the ontology.

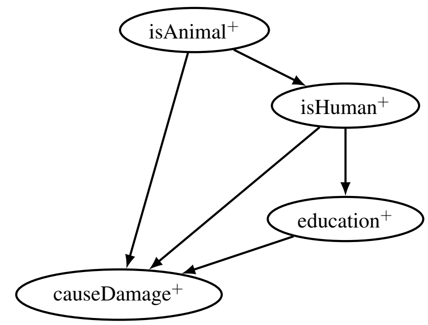

Figure 2: A belief network using the extended variants of the random variables in Example 2.

A simpler representation is to build a model with the non-extended random variables as in Figure 3. In this model, the constraints of the ontology are not represented. In particular, there is no arc from *isHuman* to *education* – the probability of *education* does not depend on *isHuman*; only the definedness of *education* does, and that is represented by the ontology. Not only do the random variables have smaller ranges, but the network is smaller (with fewer arcs).

In this paper, we address two questions: (i) how to build a model and specify probabilities and (conditional) independencies that obey the constraints implied by the underlying

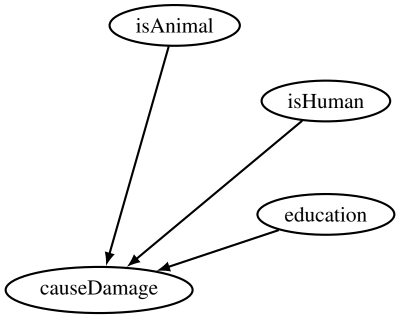

Figure 3: The graph structure of an ontologically-based belief network for Example 2.

ontology, and (ii) how to carry out probabilistic inference, by leveraging the ontology, so that undefined values do not need to be explicitly modelled in the random variables.

4 Ontologically-Based Belief Networks

In this section, we define ontologically-based belief networks (OBBNs), which represent probabilistic dependencies separately from the logical dependencies of an ontology.

4.1 Representation

Preliminary Definitions and Notation

In an OBBN, random variables are constructed from properties in the ontology[3]. For a random variable, the **domain** of its corresponding property signifies when the random variable is defined. The **range** of a random variable is the range of the corresponding property in the ontology.

Propositions are built from variable-value assignments with logical connectives. We need to be concerned with undefined values. Here, we assume pre-emptive operators, which are evaluated from left-to-right[4]:

Definition 3. A **proposition** is of one of the following forms:

- $X = x$ where X is a random variable and $x \in range(X)$, which is true if X has value x.
- $\alpha \wedge \beta$ where α and β are propositions, which is false if α is false or if α is true and β is false, and is true if both α and β are true. It is undefined if α is undefined or if α is true and β is undefined.
- $\neg\alpha$ where α is a proposition, which is true if α is false, is false if α is true, and is undefined if α is undefined.
- $\alpha \vee \beta$ where α and β are propositions, which has the same value as $\neg(\neg\alpha \wedge \neg\beta)$.

[3]Since every random variable corresponds to a property in the ontology, we use the same symbol to refer to both the variable and the property. Thus, the individuals are left implicit. The context will make clear to which of the two constructs the symbol is referring.

[4]These should be familiar to programmers, as most programming languages have non-commutative logical operations, allowing $x \neq 0 \wedge y = 1/x$ to be false when $x = 0$, as opposed to $y = 1/x \wedge x \neq 0$, which is undefined (giving an error) when $x = 0$.

These operators strictly extend the standard definitions. It is straightforward to check that, while non-commutative, the defined logical conjunction and disjunction operators remain associative. In the rest of this paper, when dealing with non-extended random variables, the notations $\wedge$, $\neg$, and $\vee$ are used to represent the non-commutative logic, instead of the commonly used operators in classical logic.

Definition 4. A proposition is **well-defined** in context *Con*, where *Con* is a proposition, with respect to an ontology if

- the proposition is of the form $X = x$, and the ontology entails that *Con* implies $dom(X)$; or
- the proposition is of the form $\alpha \wedge \beta$, α is well-defined in *Con*, and β is well-defined in $Con \wedge \alpha$; or
- the proposition is of the form $\neg\alpha$, and α is well-defined in *Con*; or
- the proposition is of the form $\alpha \vee \beta$, α is well-defined in *Con*, and β is well-defined in $Con \wedge \neg\alpha$.

A well-defined proposition (in some context *Con*) is never evaluated to *undefined*. When a context is not specified, we assume the context *True*.

We use the notation $vars(\alpha)$, for some proposition α, to denote the set of random variables that appear in α. Furthermore, for any proposition α of variable assignments, we use α^+ to denote the corresponding proposition with variables $X_i \in vars(\alpha)$ replaced by their extended variants X_i^+ but assigned the same value as X_i in α.

Structural and Probabilistic Constraints

An OBBN, like a belief network, is specified in terms of conditional probability distributions (CPDs) of random variables. We start with a total ordering of random variables $X_1, \ldots, X_n$ that is consistent with the partial order of the properties in the ontology. For each random variable X_i, a set of CPDs given the values of its parent variables, $\boldsymbol{Pa}(X_i) \subseteq \{X_1, \ldots, X_{i-1}\}$, are specified such that $P(X_i \mid X_1, \ldots, X_{i-1}) = P(X_i \mid \boldsymbol{Pa}(X_i))$.

A **parent context** [Poole and Zhang, 2003] for a random variable X is a proposition c such that $vars(c) \subseteq \boldsymbol{Pa}(X)$ and c is well-defined in the context $dom(X)$. For every random variable X, assume there is a covering set of parent contexts Π_X, such that the parent contexts in Π_X are mutually exclusive and their disjunction is implied by $dom(X)$. For every $c_i \in \Pi_X$, there is a probability distribution p_i over $range(X)$, which represents $P(X \mid c_i)$.

This construction implies that for any pair of random variables X and $Y \in vars(dom(X))$, Y is not a descendent of X. A random variable Y, whose instantiation may possibly render X undefined, cannot thus be probabilistically influenced by the value of X. This follows from the constraint that the total ordering of the random variables used in the construction of the probabilistic model is consistent with the property ordering of the ontology.

Note that the random variables involved in the primitive form of the domain of a random variable X, $vars(dom(X))$, need not be parents of X in an OBBN, unless X is also probabilistically dependent on these random variables. The CPDs of X are simply irrelevant in the cases when its domain does not hold.

Definition 5. An **ontologically-based belief network (OBBN)** is described by a 5-tuple $\langle Ont, \boldsymbol{X}, \prec, \boldsymbol{Pa}, \boldsymbol{P} \rangle$, where

- Ont is an ontology;
- $\boldsymbol{X} \subseteq Ont.\boldsymbol{Prop}$ is a set of random variables;
- $\prec$ is a total ordering of the variables that is consistent with the property ordering of Ont;
- $\boldsymbol{Pa} : \boldsymbol{X} \to \boldsymbol{2^X}$ specifies, for each $X_i \in \boldsymbol{X}$, the set of its parent variables $\boldsymbol{Pa}(X_i) \subseteq \{X \in \boldsymbol{X} : X \prec X_i\}$; and
- $\boldsymbol{P}$ is a mapping from $\boldsymbol{X}$, that maps $X \in \boldsymbol{X}$ into $\{\langle c_i, p_i \rangle : c_i \in \Pi_X\}$. Thus, $\boldsymbol{P}(X)$ specifies a conditional probability distribution for X given any proposition that is consistent with $dom(X)$.

While the definition of an OBBN does not require that the CPDs of the random variables be specified in any particular form, a natural representation of the CPDs is a tree structure, like that for context-specific independence [Boutilier *et al.*, 1996]. In the tree-structured representation of the CPDs for X, the internal nodes correspond to the parent variables, and the arcs from a node correspond to values for the random variable at that node. The domain of X conjoined with the path to a node must imply the domain of the random variable at that node in the ontology. Π_X corresponds to the set of paths down the tree, with the corresponding conditional probability distributions at the leaves.

Example 3. Figure 1 depicts a tree-structured representation of the CPDs for *causeDamage*. Note that the path $isAnimal = false$ cannot split on any other parent variable because any more splitting results in undefined conjunctions of variable assignments, and similarly for the path $isAnimal = true \wedge isHuman = false$.

Example 4. The tree-structured CPDs for *education* consist of only one single distribution (e.g., $(0.2, 0.8)$) over $\{low, high\}$, since *education* does not need a parent variable in an OBBN. We will only use *education* in the contexts where the individual under consideration is a human.

Given an OBBN $\mathcal{M} = \langle Ont, \mathbf{X}, \prec, \boldsymbol{Pa}, \boldsymbol{P} \rangle$, the corresponding **extended belief network (EBN)** $\mathcal{M}^+$ has random variables X^+ for $X \in \mathbf{X}$, and the parents of X^+ are the parents of X together with those random variables in the minimal subset $\boldsymbol{Y}$ of $vars(dom(X))$ such that for every random variable $Y \in vars(dom(X))$, either $Y \in \boldsymbol{Y}$ or $\mathcal{M}.Ont \models dom(X)_{\boldsymbol{Y}} \to dom(X)_{\{Y\}}$, where $dom(X)_{\boldsymbol{Y}}$ denotes the part of $dom(X)$ that involves the variables in $\boldsymbol{Y}$ and similarly for $dom(X)_{\{Y\}}$. In the worst case, $\boldsymbol{Y} = vars(dom(X))$. For contexts c_i where $dom(X)$ is true, $P_{\mathcal{M}^+}(X^+ \mid c_i^+) = P_{\mathcal{M}}(X \mid c_i)$, (hence $P_{\mathcal{M}^+}(X^+ = \perp \mid c_i^+) = 0$). For contexts c_i where $dom(X)$ is false, $P_{\mathcal{M}^+}(X^+ = \perp \mid c_i^+) = 1$.

4.2 Semantics

The standard semantics for probability is in terms of possible worlds, which correspond to assignments to all of the random variables. A joint probability distribution is a distribution over the possible worlds, and probabilities of individual propositions can be defined in terms of this. When there is an underlying ontology, an assignment of a value to each random variable is often undefined. If we only consider the well-defined assignments, we can get a smaller possible world structure. The analogous notion to an assignment of a value to each random variable is a maximal well-defined conjunction of variable assignments:

Definition 6. A well-defined conjunction of variable assignments $X_1 = x_1 \wedge \cdots \wedge X_k = x_k$ is **maximal** if for any variable $X' \notin \{X_1, \ldots, X_k\}$ and any value $x' \in range(X')$, $X_1 = x_1 \wedge \cdots \wedge X_k = x_k \wedge X' = x'$ is not well-defined.

To avoid considering equivalent conjunctions that only differ in the order of variable assignments, we assume that variables in a conjunction follow a total ordering consistent with the ordering of the ontology. For conjunction α, we use $\alpha_{<k}$ to denote the preceding part of α that contains all variables $X_i \in vars(\alpha)$ with $i < k$.

Whereas an EBN encodes a joint probability distribution over all conjunctions of extended variable assignments that involve all the random variables, an OBBN represents a joint probability distribution only over maximal well-defined conjunctions of the non-extended random variables. For an OBBN $\mathcal{M}$ with the total ordering $X_1, \ldots, X_n$ of random variables, the probability of a maximal well-defined conjunction, $\alpha \equiv X_{\pi(1)} = x_{\pi(1)} \wedge \cdots \wedge X_{\pi(k)} = x_{\pi(k)}$, (where π is a mapping from $\{1, \ldots, k\}$ to $\{1, \ldots, n\}$ with $\pi(i) < \pi(i+1)$), is computed as:

$$P_{\mathcal{M}}(\alpha) = \prod_{i=1}^{k} P_{\mathcal{M}}(X_{\pi(i)} = x_{\pi(i)} \mid c_{\pi(i)}),$$

where $c_{\pi(i)}$ is the parent context for $X_{\pi(i)}$ such that $\alpha_{<\pi(i)} \models c_{\pi(i)}$. By not representing the undefined worlds, an OBBN can reduce the size of the set of possible worlds by an exponential factor.

Example 5. For the EBN in Example 2, there are 4 random variables, each with 3 values (including $\perp$), and so there are $3^4 = 81$ possible worlds. The possible worlds for the OBBN for Example 3 are shown in Figure 4. Out of the 81 possible worlds for the EBN, only 8 have a non-zero probability. The others are impossible due to the ontology. There is an isomorphism between the non-zero worlds and the worlds for the EBN in Figure 4.

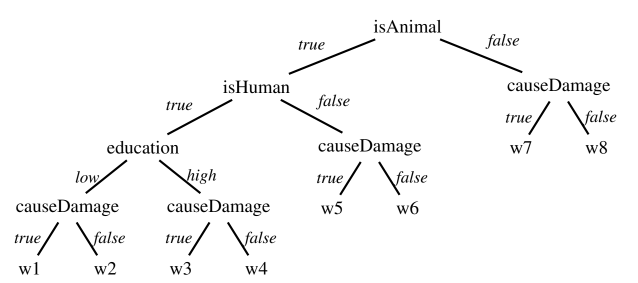

Figure 4: A tree-structured representation of all possible worlds that are encoded by an OBBN for Example 2. Every path from the root to a leaf corresponds to a maximal well-defined conjunction of variable assignments.

5 Inference

The problem of inference is to determine the posterior distribution $P(Q^+ \mid \mathcal{E})$ for a random variable Q, given some well-defined observed evidence $\mathcal{E}$. We compare three approaches:

EBN-Q Compute $P(Q^+ \mid \mathcal{E}^+)$ using the EBN.

3Q Compute $P(dom(Q) \mid \mathcal{E})$ and $P(Q \mid \mathcal{E} \wedge dom(Q))$ using the OBBN.

MQ Replace Q in the OBBN with Q^+, add explicit dependencies from $vars(dom(Q))$ to Q^+, and then compute $P(Q^+ \mid \mathcal{E})$ using the modified OBBN.

We first describe 3Q. The inference starts with querying whether the ontology entails that Q is always, never, or sometimes well-defined given $\mathcal{E}$. This relatively cheap computation can simplify or even make redundant the subsequent probabilistic queries. There are three possible outcomes for the ontological query:

1. The ontology entails that $\mathcal{E}$ implies $dom(Q)$ is false:
$$Ont \models \mathcal{E} \rightarrow \neg dom(Q).$$
The answer to the query is then: $P(Q^+ = \perp) = 1$. No probabilistic inference is needed.
2. The ontology entails that $\mathcal{E}$ implies $dom(Q)$ is true:
$$Ont \models \mathcal{E} \rightarrow dom(Q).$$
In this case, $P(Q^+ = \perp) = 0$, and for any other value $q \neq \perp$, $P(Q^+ = q \mid \mathcal{E})$ is the same as $P(Q = q \mid \mathcal{E})$ in the OBBN, (which does not contain undefined values).
3. Otherwise, it remains uncertain whether an assignment $Q = q$, for any $q \in range(Q)$, is well-defined given $\mathcal{E}$. We proceed with two separate probabilistic inference tasks, which can be run in parallel: (i) determining the probability that $Q = q$ is well-defined given $\mathcal{E}$. Let $\gamma = P(dom(Q) \mid \mathcal{E})$; and (ii) calculating the distribution of Q when it is well-defined, i.e., $P(Q \mid \mathcal{E} \wedge dom(Q))$. We return $P(Q^+ = \perp \mid \mathcal{E}) = 1 - \gamma$, and for $q \neq \perp$, $P(Q^+ = q \mid \mathcal{E}) = \gamma P(Q = q \mid \mathcal{E} \wedge dom(Q))$. Since $dom(Q)$ is well-defined, (i) needs not deal with the potential issue of undefinedness. Similarly, (ii) incorporates $dom(Q)$ into the evidence and becomes an instance of Case 2, where the query variable is always defined.

Leveraging the underlying ontology, the entire inference process of 3Q boils down to an ontological query, followed by up to two probabilistic queries. The ontological query is assumed to be a relatively cheaper computation compared to the probabilistic queries, yet in some cases, it lets us skip one or both of the probabilistic queries. This inference scheme is summarized in Procedure 3Q-INFERENCE.

Compared with 3Q, MQ allows for a single probabilistic query and so avoids repeated computation, at the expense of needing to dynamically modify the OBBN structure for each query.

6 Equivalence of Representations

In this section, we present the coherence of an OBBN and the validity of 3Q-INFERENCE by comparing it to inference using the corresponding EBN.

Procedure 3Q-INFERENCE

```
Input: OBBN M, query variable Q, and evidence E.
Output: posterior distribution P(Q+ | E), where
        range(Q+) = range(Q) ∪ {⊥}.
if Ont ⊨ E → ¬dom(Q) then
    return P(Q+ = ⊥ | E) = 1
else if Ont ⊨ E → dom(Q) then
    D ← P(Q | E)
    return {P(Q | E) = D, P(Q+ = ⊥ | E) = 0}
else
    γ ← P(dom(Q) | E)
    D ← P(Q | E ∧ dom(Q))
    return {P(Q | E) = γ·D, P(Q+ = ⊥ | E) = 1 − γ}
end
```

Theorem 1. *Every maximal well-defined conjunction $\mathcal{S}$ of variable assignments defined in an OBBN $\mathcal{M}$ can be extended to a possible world $\mathcal{S}^+_{full}$ in the corresponding EBN by adding an assignment of $\perp$ to every random variable $X \notin vars(\mathcal{S})$, and $P_{\mathcal{M}}(\mathcal{S}) = P_{\mathcal{M}^+}(\mathcal{S}^+_{full})$. All other possible worlds in $\mathcal{M}^+$ have probability* 0.

For a proof of this and other theorems, see the online appendix[5]. This shows that an OBBN represents a probability distribution over all maximal well-defined conjunctions of variable assignments.

This result is surprising because although some propositions may not be well-defined, as long as we follow the structure of an OBBN and never have a defined variable depend on an undefined variable, we never need to explicitly represent the value "undefined".

The coherence of an OBBN and the correspondence between the OBBN and the EBN directly lead to the consistency of probabilistic query results between the two models.

Theorem 2. *Let $\mathcal{M}$ be an OBBN with a total ordering $X_1, \ldots, X_n$ of random variables and $\mathcal{M}^+$ be the corresponding EBN. Let $\mathcal{S}$ be a well-defined conjunction of variable assignments. Then $P_{\mathcal{M}}(\mathcal{S}) = P_{\mathcal{M}^+}(\mathcal{S}^+)$.*

Theorem 3. *Let $\mathcal{M}$ be an OBBN and $\mathcal{M}^+$ be the corresponding EBN. For any random variable Q and well-defined evidence $\mathcal{E}$ with $P_{\mathcal{M}}(\mathcal{E}) > 0$, the posterior $P_{\mathcal{M}}(Q^+ \mid \mathcal{E})$ returned by* 3Q-INFERENCE *is the same as $P_{\mathcal{M}^+}(Q^+ \mid \mathcal{E}^+)$.*

7 Empirical Comparison

We construct an ontology that expands parametrically with the number of properties. For each property, there can be classes that are defined when the property is true, classes defined when the property is false, and classes that are defined when the property is defined but do not depend on its value. We construct one instance of each of these classes in a recursive tree structure, which we call a TAF structure (for "true, always, false"), in a breadth-first manner.

Figure 5 provides a pictorial illustration of the synthetic ontology. The range of each property is $\{true, false\}$. Prop-

[5] The appendix can be accessed at `http://www.cs.ubc.ca/~poole/papers/kuo-ijcai2013-appendix.pdf`.

erty p_0 is always defined, (e.g., the domain of p_0 could be *Thing* in an OWL ontology). Property p_1 is only defined when $p_0 = true$; p_2 is always defined regardless of the value p_0 takes; and p_3 is defined when $p_0 = false$. p_4 is defined only when $p_1 = true$, (so $dom(p_4) \equiv p_0 = true \wedge p_1 = true$). Note that although p_5 is defined no matter what value p_1 takes, it still requires that p_1 be defined, which means $p_0 = true$. Other properties are constructed in the same fashion.

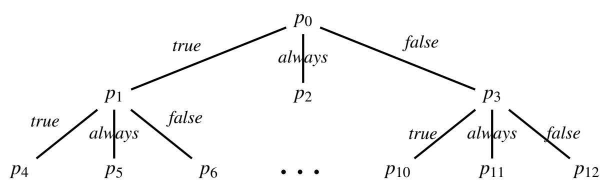

Figure 5: A pictorial visualization of a TAF-structured ontology with 13 properties, namely $p_0, p_1, \ldots, p_{12}$.

We build an OBBN using the ontology such that each random variable p_i probabilistically depends on every predecessor variable whenever it is defined, except for those variables whose values are determined when p_i is defined. As an example, for the ontology in Figure 5, p_{12} has probabilistic dependencies on p_2, p_7, p_8, p_9, and p_{11}. We query the final random variable.

We use an open-source implementation of the CVE algorithm [Poole and Zhang, 2003], which uses a compact representation for the CPDs to exploit context-specific independence. Figure 6 presents a comparison of the running times for probabilistic queries using the OBBN and its corresponding EBN. These running times are the averages of those of 500 runs of the same queries for $15 \leq n \leq 30$, 100 runs for $31 \leq n \leq 45$, and 10 runs for $46 \leq n$, where n is the number of random variables[6]. The significant difference in the running times demonstrates that inference with OBBNs can be orders of magnitude more efficient[7].

All of the measured running times are for a Java implementation running on a Windows 7 machine with an Intel dual-core i5-2520 processor and 16 GB of RAM, with up to 12 GB allocated to the Java Virtual Machine. Garbage collector was run using a separate parallel thread.

8 Conclusion

We have presented OBBNs, a simple framework that integrates belief networks with ontologies and avoids explicit modelling of undefined values in the random variables. We showed that, by exploiting the determinism that arises from the ontology, probabilistic reasoning involving variables whose corresponding properties may not be well-defined can still be effectively carried out. The proposed framework frees domain modellers from having to think about undefined values by separating the probabilistic dependencies from the logical dependencies implied by the ontology. Importantly, it also facilitates more efficient inference by leveraging the ontology to reduce the probabilistic network structures and the ranges of the random variables.

[6]For the small models, we could not get meaningful measures of the running times with one single run of each query, due to the issue of recorded numerical precision. The inference procedures are deterministic, and the variation in our implementation's running times is not significant.

[7]In our experiments, we ran the two probabilistic queries sequentially. However, they could easily be executed in parallel.

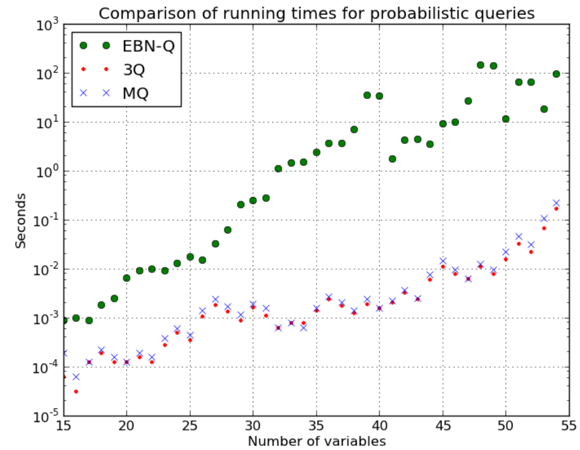

Figure 6: A comparison of running times for probabilistic queries with an OBBN and its corresponding EBN. The x-axis is linearly spaced, whereas the y-axis is spaced using the log scale.

Acknowledgement

We thank Matthew Dirks and the anonymous reviewers for their comments. This work was supported in part by NSERC, Mitacs, and the Institute for Computing, Information and Cognitive Systems (ICICS) at UBC.

References

[Aristotle, 350 BC] Aristotle. *Categories*. Translated by E. M. Edghill, 350 B.C.

[Berg, 1982] Jan Berg. Aristotle's theory of definition. In *ATTI del Convegno Internazionale di Storia della Logica*, pages 19–30, San Gimignano, 1982.

[Boutilier *et al.*, 1996] Craig Boutilier, Nir Friedman, Moises Goldszmidt, and Daphne Koller. Context-specific independence in Bayesian networks. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI-96)*, pages 115–123, 1996.

[Darwiche, 2001] Adnan Darwiche. Recursive conditioning. *Artificial Intelligence*, 126(1-2):5–41, 2001.

[Darwiche, 2009] Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

[De Raedt *et al.*, 2008] Luc De Raedt, Paolo Frasconi, Kristian Kersting, and Stephen H. Muggleton, editors. *Probabilistic Inductive Logic Programming*. Springer, 2008.

[Dechter, 1996] R. Dechter. Bucket elimination: A unifying framework for probabilistic inference. In E. Horvitz and F. Jensen, editors, *UAI-96*, pages 211–219, Portland, OR, 1996.

[Getoor and Taskar, 2007] Lise Getoor and Ben Taskar, editors. *Introduction to Statistical Relational Learning*. MIT Press, Cambridge, MA, 2007.

[Grau *et al.*, 2008] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter Patel-Schneider, and Ulrike Sattler. Owl 2: The next step for owl. *Web Semant.*, 6(4):309–322, November 2008.

[Koller and Friedman, 2009] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009.

[Motik *et al.*, 2012a] Boris Motik, Peter F. Patel-Schneider, and Bernardo Cuenca Grau. Owl 2 web ontology language: Direct semantics (second edition). *W3C Recommendation, 11 December 2012*, 2012.

[Motik *et al.*, 2012b] Boris Motik, Peter F. Patel-Schneider, and Bijan Parsia. Owl 2 web ontology language: Structural specification and functional-style syntax (second edition). *W3C Recommendation, 11 December 2012*, 11 December 2012.

[Pearl, 1988] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[Poole and Zhang, 2003] David Poole and Nevin Lianwen Zhang. Exploiting contextual independence in probabilistic inference. *Journal of Artificial Intelligence Research*, 18:263–313, 2003. Code is available from `http://www.cs.ubc.ca/~poole/code.html`.

[Poole *et al.*, 2008] David Poole, Clinton Smyth, and Rita Sharma. Semantic science: Ontologies, data and probabilistic theories. In *Uncertainty Reasoning for the Semantic Web I*, LNAI/LNCS. Springer, 2008.

[Poole *et al.*, 2009] David Poole, Clinton Smyth, and Rita Sharma. Ontology design for scientific theories that make probabilistic predictions. *IEEE Intelligent Systems*, 24(1):27–36, Jan/Feb 2009.

[Smith, 2003] Barry Smith. Ontology. In L. Floridi, editor, *Blackwell Guide to the Philosophy of Computing and Information*, pages 155–166. Oxford: Blackwell, 2003.

[Zhang and Poole, 1994] N. L. Zhang and D. Poole. A simple approach to Bayesian network computations. In *Proc. 10th Canadian Conference on AI*, pages 171–178, 1994.

[Zhang and Poole, 1999] N. L. Zhang and D. Poole. On the role of context-specific independence in probabilistic reasoning. In *IJCAI-99*, pages 1288–1293, Stockholm, Sweden, August 1999.