A Bayesian Approach to Uncertainty Modelling in OWL Ontology

Zhongli DING, Yun PENG, Rong PAN

***Abstract*-- Dealing with uncertainty is crucial in ontology engineering tasks such as domain modelling, ontology reasoning, and concept mapping between ontologies. This paper presents our on-going research on modelling uncertainty in ontologies based on Bayesian networks (BN). This includes 1) extending OWL to allow additional probabilistic markups for attaching probability information, 2) directly converting a probabilistically annotated OWL ontology into a BN structure by a set of structural translation rules, and 3) constructing the conditional probability tables (CPTs) of this BN using a new method based on iterative proportiobal fitting procedure (IPFP). The translated BN can support more accurate ontology reasoning under uncertainty as Bayesian inferences.**

***Index Terms*-- Bayesian Networks, IPFP, Ontology, Semantic Web, Uncertainty.**

I. Introduction and Motivation

In the semantic web [17], an important component of an ontology defined in OWL [18] or RDF(S) [19] is the taxonomical concept subsumption hierarchy based on class axioms (defined by rdfs:subClassOf, owl:equivalentClass, and owl:disjointWith) and logical relations among the concept classes (defined by owl:unionOf, owl:intersectionOf, and owl:complementOf). Such an ontology taxonomy definition is based on crisp logic and thus cannot quantify the degree of the overlap or inclusion between two concepts, cannot support reasoning in how close a description D is to its most specific subsumer and most general subsumee, and tends to over-generalize with noisy input [2]. Uncertainty becomes more prevalent in web environment when more than one ontology are involved where it is often the case that a concept defined in one ontology can only find partial matches to one or more concepts in another ontology.

To model uncertainty in ontology representation, reasoning and mapping, this paper presents a new probabilistic extension to OWL ontology taxonomy based on Bayesian networks (BN) [1], a widely used graphic model of dependencies among variables. In our approach, OWL is first augmented to allow additional probabilistic markups so that probability values can be attached to individual concepts in an ontology. Secondly, a set of structural translation rules is defined to convert this probabilistically annotated OWL ontology taxonomy into a directed acyclic graph (DAG) of a BN. Finally, the BN is completed by constructing conditional probability tables (CPTs) for each node in the DAG.

To help understand our approach, in the remaining of this section, we give a simple review of OWL [18] and BN [1].

A. Web Ontology Language (OWL)

An OWL document can include an optional ontology header and any number of classes, properties, axioms, and individual descriptions. In an ontology defined by OWL, a named class is described by a class identifier. An anonymous class can be described by some value (owl:hasValue, owl:allValuesFrom, owl:someValuesFrom) or cardinality (owl:cardinality, owl:-maxCardinality, owl:minCardinality) restriction on property (owl:Restriction); by exhaustively enumerating all the individuals that form the instances of this class (owl:oneOf); or by logical operation on two or more classes (owl:unionOf, owl:intersectionOf, owl:complementOf). Three class axioms (rdfs:subClassOf, owl:equivalentClass, owl:disjointWith) can be used for defining necessary and sufficient conditions of a class. Two kinds of properties can be defined: object property (owl:ObjectProperty) which links individuals to individuals, and datatype property (owl:DatatypeProperty) which links individuals to data values. "rdfs:subPropertyOf" is used to define that one property is a subproperty of another property. Besides these most commonly used constructors, there are some other constructors (e.g., owl:equivalentProperty and owl:inverseOf to relate two properties; owl:FunctionalProperty and owl:InverseFunctionalProperty to impose cardinality restrictions on properties; etc.)

The semantics of OWL is defined based on model theory in the way analogous to the semantics of description logic (DL). With a set of vocabulary (mostly as described above), one can define an ontology as a set of (restricted) RDF triples which can be represented as a RDF graph.

B. Bayesian Network

In the most general form, a BN of n variables consists of a DAG of n nodes and a number of arcs. Nodes X_i in a DAG correspond to random variables, and directed arcs between two nodes represent direct causal or influential relations from one

Zhongli Ding is with Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore, Maryland 21250 USA (telephone: 1-410-455-3971, e-mail: zding1@ cs.umbc.edu).

Dr. Yun Peng is with Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore, Maryland 21250 USA (telephone: 1-410-455-3816, fax: 1-410-455-3969, e-mail: ypeng@ cs.umbc.edu).

Rong Pan is with Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore, Maryland 21250 USA (telephone: 1-410-455-3971, e-mail: panrong1@ cs.umbc.edu).

variable to the other. The uncertainty of the causal relationship is represented locally by the CPT $P(X_i \mid \pi_i)$ associated with each node X_i, where π_i is the parent set of X_i. Under a conditional independence assumption, the joint probability distribution of $X = (X_1, ..., X_n)$ can be factored out as a product of the CPTs in the network (named "the chain rule of BN"): $P(X = x) = \prod_{i=1}^{n} P(x_i \mid \pi_i)$. With the joint probability distribution, BN supports, at least in theory, any probabilistic inference in the joint space.

Besides the power of probabilistic reasoning provided by BN itself, we are attracted to BN in this work also by the structural similarity between the DAG of a BN and the RDF graph of OWL ontology: both of them are directed graphs, and direct correspondence exists between many nodes and arcs in the two graphs. In this work, we only consider ontology taxonomy which uses only constructors for the terminology part of DL. Constructors related to properties, individuals, and datatypes will be considered in the future.

The rest of this paper is organized as follows: Section II extends OWL for encoding probabilities into ontology; Section III presents a set of rules that are used to translate OWL ontology into DAG of BN; Section IV develops a method to construct CPTs for each node in the DAG; Section V briefly discusses how ontology reasoning may be performed over this translated BN. The paper concludes in Section VI with discussions of related work and directions for future research.

II. Encoding Probabilities in Ontology

The model-theoretic semantics of OWL [18] treats the domain as a non-empty collection of individuals. If classes A and B represent two concepts, we treat them as random binary variables and interpret $P(A = a)$ as the prior probability or one's belief that an arbitrary individual belongs to class A, and $P(a \mid b)$ as the conditional probability that an individual of class B also belongs to class A. Similarly, we can interpret $P(\bar{a})$, $P(\bar{a} \mid b)$, $P(a \mid \bar{b})$, and $P(\bar{a} \mid \bar{b})$ with the negation interpreted as "not belonging to". These two types of probabilities (prior or conditional) correspond naturally to classes and relations in an ontology, and are most likely to be available to ontology designers. Currently, our translation framework can encode two types of probabilistic information into the original ontology: for a concept class C and its parent superconcept class set π_C

(1) Prior or marginal probability $P(C)$;

(2) Conditional probability $P(C \mid O_C)$ where $O_C \subseteq \pi_C$, $\pi_C \neq \varnothing$, $O_C \neq \varnothing$.

To add such uncertainty information into an existing ontology, we treat a probability as a kind of resource, and define two OWL classes: "PriorProb", "CondProb". A probability with the form $P(C)$ is defined as an instance of class "PriorProb", which has two mandatory properties: "hasVarible" and "hasProbValue". A probability with the form $P(C \mid O_C)$ is defined as an instance of class "CondProb" with three mandatory properties: "hasCondition" (at least one), "hasVariable", and "hasProbValue". The range of properties "hasCondition" and "hasVariable" is a defined class named "Variable", which has two mandatory properties: "hasClass" and "hasState". "hasClass" points to the concept class this probability is about and "hasState" gives the "True" (belong to) or "False" (not belong to) state of this probability.

For example, $P(c) = 0.8$, the prior probability that an arbitrary individual belongs to class C, can be expressed as follows:

```
<Variable rdf:ID="c">
  <hasClass>C</hasClass>
  <hasState>True</hasState>
</Variable>
<PriorProb rdf:ID="P(c)">
  <hasVariable>c</hasVariable>
  <hasProbValue>0.8</hasProbValue>
</PriorProb>
```

and $P(c \mid p1, p2, p3) = 0.8$, the conditional probability that an individual of the intersection class of $P1$, $P2$, and $P3$ also belongs to class C, can be expressed as follows:

```
<Variable rdf:ID="c">
  <hasClass>C</hasClass>
  <hasState>True</hasState>
</Variable>
<Variable rdf:ID="p1">
  <hasClass>P1</hasClass>
  <hasState>True</hasState>
</Variable>
<Variable rdf:ID="p2">
  <hasClass>P2</hasClass>
  <hasState>True</hasState>
</Variable>
<Variable rdf:ID="p3">
  <hasClass>P3</hasClass>
  <hasState>True</hasState>
</Variable>
<CondProb rdf:ID="P(c|p1, p2, p3)">
  <hasCondition>p1</hasCondition>
  <hasCondition>p2</hasCondition>
  <hasCondition>p3</hasCondition>
  <hasVariable>c</hasVariable>
  <hasProbValue>0.8</hasProbValue>
</CondProb>
```

For simplicity we did not consider the namespaces in above examples. For a complete definition of probabilistic markups, please refer to: http://www.csee.umbc.edu/~zding1/owl/prob.owl.

III. Structural Translation

The ontology augmented with probability values as described in Section II will still be an OWL file. It can be translated into a BN by first forming a DAG following a set of rules. The general principle underlying these rules is that all classes (specified as "subjects" and "objects" in RDF triples of the OWL file) are translated into nodes in BN, and an arc is drawn between two nodes in BN if the corresponding two classes are related by a "predicate" in the OWL file, with the direction from the superclass to the subclass if it can be determined. Control nodes are created during the translation to facilitate modelling relations among class nodes that are related by OWL logical operator. These structural translation rules are summarized as follows:

(1) Every primitive or defined concept class C, is mapped into a two-state (either "True" or "False") variable node in the translated BN, C is in "True" state when an instance c belongs to it;

(2) There is a directed arc from a parent superclass node to a subclass node, for example, a concept class C defined with superconcept classes C_i ($i = 1, ..., n$) by "rdfs:subClassOf" is mapped into a subnet in the translated BN with one converging connection (Fig.1) from each C_i to C;

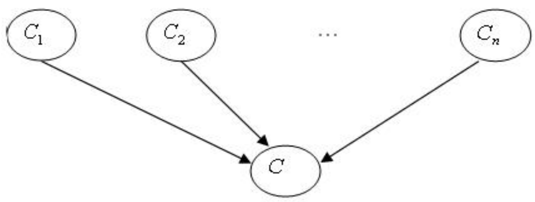

Fig.1. – "rdfs:subClassOf"

(3) A concept class C defined by set intersection operation (owl:intersectionOf) of concept classes C_i ($i = 1, ..., n$) is mapped into a subnet (Fig.2) in the translated BN with one converging connection from each C_i to C, and one converging connection from C and each C_i to a control node called "Bridge_Intersection";

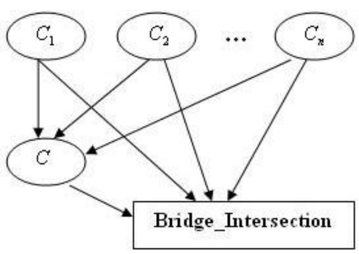

Fig.2. – "owl:intersectionOf"

(4) A concept class C defined by set union operation (owl:unionOf) of concept classes C_i ($i = 1, ..., n$) is mapped into a subnet (Fig.3) in the translated BN with one converging connection from C to each C_i, and one converging connection from C and each C_i to a control node called "Bridge_Union";

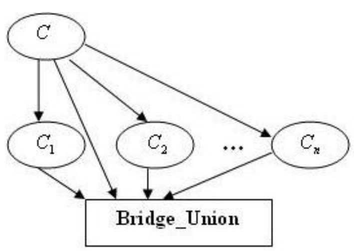

Fig.3. – "owl:unionOf"

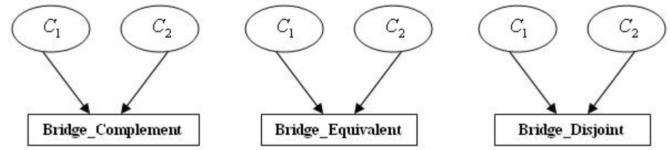

Fig.4. – "owl:complementOf, owl:equivalentClass, owl:disjointWith"

(5) If two concept classes C_1 and C_2 are related by complement (owl:complementOf), equivalent (owl:equivalent-Class), or disjoint (owl:disjointWith) relation, then a control node (named "Bridge_Complement", "Bridge_Equivalent", "Bridge_Disjoint" respectively, as in Fig.4) is added to the translated BN, and there are directed links from C_1 and C_2 to this node.

Based on rule (1) to (5), the translated BN contains two kinds of nodes: regular nodes for concept classes and control nodes which bridging nodes that are associated by logical relations. The CPT of a control node will be set in a way such that when the state of this control node is set to "True", the corresponding logical relation among its parent concept class nodes will be held (see Subsection IV.A for more details). By using control nodes, the logical relations are separated from the "rdfs:subClassOf" relation, so the in-arcs to a regular node C will only come from its parent superclass nodes, which makes C's CPT smaller and easier to construct, compared to our old method in [2]. In the translated BN, all the arcs are directed based on OWL statements, two concept class nodes without any defined or derived relations are d-separated with each other, and two implicitly dependent concept class nodes are d-connected with each other but there is no arc between them.

IV. CONSTRUCTING CPTS

Once we had the network structured, the last step to complete the translation is to assign a conditional probability table (CPT) $P(C \mid \pi_C)$ to each variable node C in the structure, where π_C is the set of all parent nodes of C. From structural translation we know that all nodes X in the translated BN can be partitioned into two subsets: regular nodes X_R which denote concept classes, and control nodes X_C for bridging nodes that are associated by logical relations. For a regular node $C \in X_R$, as described in Section II, we have prior probability $P(C)$ attached to it if it does not have any parent nodes; or conditional probability $P(C \mid O_C)$ attached to it if its parent set $\pi_C \neq \varnothing$ and $O_C \subseteq \pi_C$. Details about how to construct CPTs for regular nodes in X_R based on attached probabilistic information in the probabilistically annotated ontology will be given later in Subsection C. Here we deal with CPTs for the control nodes in X_C first.

A. CPTs for Control Nodes

Based on the structural translation rules, there are five types of control nodes corresponding to the five logic operators in

OWL. They are “Bridge_Complement”, “Bridge_Disjoint”, “Bridge_Equivalent”, “Bridge_Intersection”, “Bridge_Union”. Their CPTs are determined by the logical relation among its parent concept class nodes, which are specified next.

(1) Bridge_Complement (Table 1): When its state is set to “True”, C_1 and C_2 are complement of each other;

Table 1 – CPT of Bridge_Complement

C1	C2	True	False
True	True	0.000	100.00
True	False	100.00	0.000
False	True	100.00	0.000
False	False	0.000	100.00

(2) Bridge_Disjoint (Table 2): When its state is set to “True”, C_1 and C_2 are disjoint with each other;

Table 2 – CPT of Bridge_Disjoint

C1	C2	True	False
True	True	0.000	100.00
True	False	100.00	0.000
False	True	100.00	0.000
False	False	100.00	0.000

(3) Bridge_Equivalent (Table 3): When its state is set to “True”, C_1 and C_2 are equivalent with each other;

Table 3 – CPT of Bridge_Equivalent

C1	C2	True	False
True	True	100.00	0.000
True	False	0.000	100.00
False	True	0.000	100.00
False	False	100.00	0.000

(4) Bridge_Intersection (Table 4): When its state is set to “True”, C is the intersection of C_1 and C_2;

Table 4 – Bridge_Intersection

C1	C2	C	True	False
True	True	True	100.00	0.000
True	True	False	0.000	100.00
True	False	True	0.000	100.00
True	False	False	100.00	0.000
False	True	True	0.000	100.00
False	True	False	100.00	0.000
False	False	True	0.000	100.00
False	False	False	100.00	0.000

In a more general case, if a concept class C is the intersection of $n > 2$ concept classes then the 2^{n+1} entries in the CPT of “Bridge_Intersection” can be obtained analogously.

(5) Bridge_Union (Table 5): When its state is set to “True”, C is the union of C_1 and C_2;

Table 5 – Bridge_Union

C1	C2	C	True	False
True	True	True	100.00	0.000
True	True	False	0.000	100.00
True	False	True	100.00	0.000
True	False	False	0.000	100.00
False	True	True	100.00	0.000
False	True	False	0.000	100.00
False	False	True	0.000	100.00
False	False	False	100.00	0.000

In a more general case, if a concept class C is the union of $n > 2$ concept classes then the 2^{n+1} entries in the CPT of “Bridge_Union” can be obtained analogously.

When the CPTs for control nodes are properly determined as above, if we set the states of all the control nodes to “True”, the logical relations defined in the original ontology will be held in the translated BN, which is thus consistent with the OWL semantics. We denote this situation that all the control nodes in the translated BN are in “True” state as CT.

The remaining issue is to construct CPTs for the regular nodes in X_R so that $P(X_R | CT)$, the joint probability distribution of all regular nodes in the subspace of CT, is consistent with all the given prior and conditional probabilities attached to the nodes in X_R. This issue is difficult because 1) the product of CPTs of all variables gives the joint distribution in the general space, not the subspace of CT (the dependencies changes when going from the general space to the subspace of CT); and 2) the probabilistic information encoded is in the form of prior probability ($P(C)$) and conditional probability ($P(C | O_C)$, $\pi_C \neq \varnothing$, $O_C \subseteq \pi_C$), not directly in the form of CPT (C may have other parent nodes in addition to O_C).

To address these issues, we developed an algorithm to approximate these CPTs for X_R based on the “iterative proportional fitting procedure” (IPFP) [3]-[8], a well-known mathematical procedure that modifies a given distribution to meet a set of constraints while minimizing *I-divergence* (Kullback-Leibler distance) to the original distribution.

B. *Brief Introduction to IPFP*

In this subsection we give a brief introduction to the iterative proportional fitting procedure (IPFP), which was first published by [3] in 1937, and in [4] it was proposed as a procedure to estimate cell frequencies in contingency tables under some marginal constraints. In 1975, I. Csiszar [5] provided an IPFP convergence proof based on *I-divergence* geometry. J. Vomlel rewrote a discrete version of this proof in his PhD thesis [6] in 1999. IPFP was extended in [7], [8] as conditional iterative proportional fitting procedure (CIPF-P) to also take conditional distributions as constraints, and the convergence was established for the finite discrete case.

We give definitions of *I-divergence* and *I-projection* first before going into the details of IPFP. In our context, all random variables are finite and all probability distributions are discrete.

Definition 3.1 (*I-divergence*)
Let $\boldsymbol{P}$ be a set of probability distributions, and for $P, Q \in \boldsymbol{P}$, *I-divergence* (also known as *Kullback-Leibler divergence* or *Cross-entropy*, which is often used as a distance measure between two probability distributions) is defined as:

$$I(P \| Q) = \begin{cases} \sum_{x \in X, P(x)>0} P(x) \log \frac{P(x)}{Q(x)} & \text{if } P \ll Q \\ +\infty & \text{if } P \not\ll Q \end{cases} \quad (1)$$

Here $P << Q$ means P is dominated by Q, i.e.
$\{x \in X \mid P(x) > 0\} \subseteq \{y \in X \mid Q(y) > 0\}$
where x (or y) is an assignment of X, or equivalently:
$\{y \in X \mid Q(y) = 0\} \subseteq \{x \in X \mid P(x) = 0\}$
since a probability value is always non-negative. The dominance condition in (1) guarantees division by zero will not occur because whenever the denominator $Q(x)$ is zero, the numerator $P(x)$ will be zero. Note that *I-divergence* is zero if and only if P and Q are identical and *I-divergence* is non-symmetric.

Definition 3.2 (*I-projection*)
The *I₁-projection* of a probability distribution $Q \in \boldsymbol{P}$ on a set of probability distributions $\boldsymbol{\varepsilon}$ is a unique probability distribution $P \in \boldsymbol{\varepsilon}$ such that the *I-divergence* " $I(P \| Q)$ " is minimal among all probability distributions in $\boldsymbol{\varepsilon}$. Similarly, the *I₂-projections* of Q on $\boldsymbol{\varepsilon}$ are probability distributions in $\boldsymbol{\varepsilon}$ that minimize the *I-divergence* " $I(Q \| P)$ " and *I₂-projection* is not generally unique.

If $\boldsymbol{\varepsilon}$ is a given set of probability distributions that satisfies all given constraints, the *I₁-projection* $P \in \boldsymbol{\varepsilon}$ of Q is a distribution that has the minimum distance from Q among all those in $\boldsymbol{\varepsilon}$ [6].

Definition 3.3 (IPFP)
Let $X = \{X_1, X_2, ..., X_n\}$ be a space of discrete random variables, given a consistent set of m marginal probability distributions $\{R(S_i)\}$ where $X \supseteq S_i \neq \varnothing$ and an initial probability distribution $Q_{(0)} \in \boldsymbol{P}$, iterative proportional fitting procedure (IPFP) is a procedure for determining a joint distribution $P(X) = P(X_1, X_2, ..., X_n) << Q_{(0)}$ satisfying all constraints in $\{R(S_i)\}$ by repeating the following computational process over k and $i = ((k-1) \bmod m) + 1$:

$$Q_{(k)}(X) = \begin{cases} 0 & \text{if } Q_{(k-1)}(S_i) = 0 \\ Q_{(k-1)}(X) \cdot \dfrac{R(S_i)}{Q_{(k-1)}(S_i)} & \text{if } Q_{(k-1)}(S_i) > 0 \end{cases} \qquad (2)$$

This process iterates over distributions in $\{R(S_i)\}$ in cycle. It can be shown [6] that in each step k, $Q_{(k)}(X)$ is an *I₁-projection* of $Q_{(k-1)}(X)$ that satisfies the constraint $R(S_i)$, and $Q^* = \lim_{k \to \infty} Q_{(k)}$ is an *I₁-projection* of $Q_{(0)}$ satisfying all constraints, i.e., Q^* converges to $P(X) = P(X_1, X_2, ..., X_n)$.

CIPF-P from [7], [8] is an extension of IPFP to allow constraints with the form of conditional probability distributions, i.e. $R(S_i \mid L_i)$ where $L_i \subseteq X$. The procedure can be written as:

$$Q_{(k)}(X) = \begin{cases} 0 & \text{if } Q_{(k-1)}(S_i \mid L_i) = 0 \\ Q_{(k-1)}(X) \cdot \dfrac{R(S_i \mid L_i)}{Q_{(k-1)}(S_i \mid L_i)} & \text{if } Q_{(k-1)}(S_i \mid L_i) > 0 \end{cases} \qquad (3)$$

CIPF-P has similar convergence result [8] as IPFP and (2) is in fact a special case of (3) with $L_i = \varnothing$.

C. Constructing CPT for Regular Nodes

Let $X = \{X_1, ..., X_n\}$ be the set of binary (i.e. $X_i \in \{x_i, \overline{x_i}\}$) variables in the translated BN, X_R the set of regular nodes, and X_C the set of control nodes, as stated earlier in this section. The remaining issue is to construct CPTs $Q(V_i \mid \pi_{V_i})$ for the regular nodes V_i in X_R so that $Q(X_R \mid CT)$, the joint probability distribution of X_R in the subspace of CT, is consistent with all the given prior and conditional probabilities. Again, we restrict the encoded probabilities to the two forms: (1) prior or marginal probability $P(C)$ and (2) conditional probability $P(C \mid O_C)$ where $O_C \subseteq \pi_C$, $\pi_C \neq \varnothing$, $O_C \neq \varnothing$, and each is attached to a node in X_R. This is a constraint satisfaction problem in the scope of IPFP. However, it would be very expensive in each iteration of (3) to compute the joint distribution $Q_{(k)}(X)$ over all the variables and then decompose it into CPTs at the end. We provide a new algorithm (called **Decomposed-IPFP** or **D-IPFP** for short) to overcome this problem by utilizing the chain rules of BN [1].

Let $P_{init}(X) = \prod_{X_i \in X} P_{init}(X_i \mid \pi_i)$ be the initial distribution of the translated BN where CPTs for control nodes in X_C are set properly as in Subsection A and CPTs for regular nodes in X_R are set to some arbitrary values that are consistent with the semantics of the subclass relation between parent and child nodes. Let $\{R(V_i \mid L_i)\}$ be the set of m given prior ($L_i = \varnothing$) or conditional ($\pi_{V_i} \supseteq L_i \neq \varnothing$) probability distributions associated with $V_i \in X_R$. The basic idea of our approach is: in each iteration step k, instead of computing a new joint probability distribution $Q_{(k)}(X)$ over all the variables on one constraint in $\{R(V_i \mid L_i)\}$, we compute a new CPT $Q_{(k)}(V_i \mid \pi_{V_i})$ for node V_i over that constraint. The iteration process loops continuously over all $R(V_i \mid L_i)$ until Q converges. D-IPFP is given below:

$$\begin{aligned} Q_{(0)} &= P_{init}(X) = \prod_{X_i \in X} P_{init}(X_i \mid \pi_i) \\ Q_{(k)}(V_i \mid \pi_{V_i}) &= Q_{(k-1)}(V_i \mid \pi_{V_i}) \cdot \frac{R(V_i \mid L_i)}{Q_{(k-1)}(V_i \mid L_i, CT)} \cdot \alpha_{k-1}(\pi_{V_i}) \end{aligned} \qquad (4)$$

where $\alpha_{k-1}(\pi_{V_i}) = \dfrac{1}{\sum_{V_i \in \{v_i, \overline{v_i}\}} (Q_{(k-1)}(V_i \mid \pi_{V_i}) \cdot R(V_i \mid L_i) / Q_{(k-1)}(V_i \mid L_i, CT))}$
is the normalization factor for each possible value assignment of π_{V_i}.

To guarantee the dominance of $Q_{(0)}$, we define $Q_{(k)}(V_i \mid \pi_{V_i}) = 0$ if $Q_{(k-1)}(V_i \mid L_i, CT) = 0$. It can be shown that (Subsection D), if the ontology definition is consistent, given an consistent and complete input set $\{R(V_i \mid L_i)\}$, Q

converges to Q^* with $Q^*(X_R | CT)$ an *I1-projection* of $P_{init}(X_R | CT)$ over $\{R(V_i | L_i)\}$ (i.e. $Q^*(X_R | CT)$ has minimum Kullback-Leibler distance to $P_{init}(X_R | CT)$ and $\forall V_i \in X_R : Q^*(V_i | L_i, CT) = R(V_i | L_i)$).

D. Convergence Proof of D-IPFP

From previous subsections we have the set of all variables $X = X_R \cup X_C$ with $X_R \cap X_C = \varnothing$ and $X_R \neq \varnothing$, where $X_R = \{V_1,...,V_s\}$ denotes the set of binary (i.e. $V_i \in \{v_i, \overline{v_i}\}$) regular nodes, $X_C = \{B_1,...,B_t\}$ denotes the set of binary (i.e. $B_i \in \{b_i, \overline{b_i}\}$) control nodes (if $X_C \neq \varnothing$).

Probability constraints can be put in a general form of $R(V_i | L_i)$ where $L_i \subseteq \pi_{V_i}$. If $L_i = \varnothing$, then the constraint is a prior or marginal, otherwise, a conditional (given some or all parents of V_i).

By the chain rule of BN [1], the probability distribution of $X_R = \{V_j \mid j = 1,...,s\}$ in the subspace of CT is:

$$\begin{aligned} &Q_{(k)}(X_R | CT) \qquad (5)\\ &= Q_{(k)}(X_R, CT) / Q_{(k)}(CT)\\ &= Q_{(k)}(X_R \setminus \{V_i\}, V_i, CT) / Q_{(k)}(CT)\\ &= Q_{(k)}(V_i | \pi_{V_i}) \cdot \prod_{B_j \in X_C} Q_{(k)}(b_j | \pi_{B_j}) \prod_{X_j \in X_R, j \neq i} Q_{(k)}(V_j | \pi_{V_j}) / Q_{(k)}(CT) \end{aligned}$$

From (4) we have:

$$Q_{(k)}(V_i | \pi_{V_i}) = \alpha_{k-1}(\pi_{V_i}) \cdot Q_{(k-1)}(V_i | \pi_{V_i}) \cdot \frac{R(V_i | L_i)}{Q_{(k-1)}(V_i | L_i, CT)} \qquad (6)$$

Substitute (6) into (5), also note that only one table, namely $Q_{(k)}(V_i | \pi_{V_i})$, is changed at iteration k, then

$$\begin{aligned} &Q_{(k)}(X_R | CT)\\ &= [\alpha_{k-1}(\pi_{V_i}) \cdot Q_{(k-1)}(V_i | \pi_{V_i}) \cdot \frac{R(V_i | L_i)}{Q_{(k-1)}(V_i | L_i, CT)}]\\ &\quad \cdot \prod_{B_j \in X_C} Q_{(k)}(b_j | \pi_{B_j}) \prod_{X_j \in X_R, j \neq i} Q_{(k)}(V_j | \pi_{V_j}) / Q_{(k)}(CT)\\ &= [\alpha_{k-1}(\pi_{V_i}) \cdot Q_{(k-1)}(V_i | \pi_{V_i}) \cdot \frac{R(V_i | L_i)}{Q_{(k-1)}(V_i | L_i, CT)}]\\ &\quad \cdot \prod_{B_j \in X_C} Q_{(k-1)}(b_j | \pi_{B_j}) \prod_{X_j \in X_R, j \neq i} Q_{(k-1)}(V_j | \pi_{V_j}) / Q_{(k)}(CT)\\ &= \alpha_{k-1}(\pi_{V_i}) \cdot \frac{R(V_i | L_i)}{Q_{(k-1)}(V_i | L_i, CT)} \cdot Q_{(k-1)}(X_R | CT) \cdot \frac{Q_{(k-1)}(CT)}{Q_{(k)}(CT)}\\ &= \beta_{k-1}(\pi_{V_i}) \cdot \frac{R(V_i | L_i)}{Q_{(k-1)}(V_i | L_i, CT)} \cdot Q_{(k-1)}(X_R | CT) \qquad (7) \end{aligned}$$

where $\beta_{k-1}(\pi_{V_i}) = \alpha_{k-1}(\pi_{V_i}) \cdot \dfrac{Q_{(k-1)}(CT)}{Q_{(k)}(CT)}$.

Now we show that $Q_{(k)}$ converges to a limit probability distribution Q^* and Q^* fulfills all the given constraints in the subspace of CT, i.e.

$\forall V_i \; Q^*(V_i | L_i, CT) = R(V_i | L_i)$, i.e.

$$\forall V_i \lim_{k \to \infty} Q_{(k)}(V_i | L_i, CT) = R(V_i | L_i) \qquad (8)$$

First, we prove that in each iteration step of (4), $Q_{(k)}(X_R | CT)$ is an *I1-projection* of $Q_{(k-1)}(X_R | CT)$ over some constraint in the subspace of CT. Because our rule (4) is for local updates (change CPTs, not the joint distribution of X_R), and because the CPTs are given for the general space but constraints are in the subspace of CT, *I1-projection* generated at each iteration does not necessarily meet the given constraint $R(V_i | L_i)$. However, we can show that $Q_{(k)}(X_R | CT)$ is an *I1-projection* of $Q_{(k-1)}(X_R | CT)$ over another constraint derived from $R(V_i | L_i)$ in the subspace of CT.

Let $L_i' = \pi_{V_i} \setminus L_i$ (or π_{V_i} is partitioned into L_i and L_i'), we define a new constraint $R'_{(k)}(V_i | \pi_{V_i}, CT)$:

$$\begin{aligned} &R'_{(k)}(V_i | L_i, L_i', CT)\\ &= \beta_{k-1}(\pi_{V_i}) \cdot \frac{R(V_i | L_i)}{Q_{(k-1)}(V_i | L_i, CT)} \cdot Q_{(k-1)}(V_i | L_i, L_i', CT) \qquad (9) \end{aligned}$$

To prove that $Q_{(k)}(X_R | CT)$ is an *I1-projection* of $Q_{(k-1)}(X_R | CT)$ over $R'_{(k)}(V_i | \pi_{V_i}, CT)$ in the subspace of CT, from (7) and (9) we have:

$$\begin{aligned} &Q_{(k)}(X_R | CT)\\ &= \beta_{k-1}(\pi_{V_i}) \cdot \frac{R(V_i | L_i)}{Q_{(k-1)}(V_i | L_i, CT)} \cdot Q_{(k-1)}(X_R | CT) \cdot \frac{Q_{(k-1)}(V_i | L_i, L_i', CT)}{Q_{(k-1)}(V_i | L_i, L_i', CT)}\\ &= Q_{(k-1)}(X_R | CT) \cdot \frac{R'_{(k)}(V_i | L_i, L_i', CT)}{Q_{(k-1)}(V_i | L_i, L_i', CT)}\\ &= Q_{(k-1)}(X_R | CT) \cdot \frac{R'_{(k)}(V_i | \pi_{V_i}, CT)}{Q_{(k-1)}(V_i | \pi_{V_i}, CT)} \qquad (10) \end{aligned}$$

Then from (3), $Q_{(k)}(X_R | CT)$ is an *I1-projection* of $Q_{(k-1)}(X_R | CT)$ over constraint $R'_{(k)}(V_i | \pi_{V_i}, CT)$ in the subspace of CT, and thus

$$Q_{(k)}(V_i | \pi_{V_i}, CT) = R'_{(k)}(V_i | \pi_{V_i}, CT) \qquad (11)$$

Second, since each iteration is an *I1-projection*, we can show (analogous to the convergence proof in [6] (Page 22)) that:

$$I(Q_{(k)}(X_R | CT) \| Q_{(k-1)}(X_R | CT)) \to 0 \qquad (12)$$

and since all the random variables are finite, based on Theorem 2.4 of J. Vomlel's thesis [6] (Page 20) and (12), the sequence $Q_{(0)}, Q_{(1)}, ..., Q_{(k-1)}, Q_{(k)}, ...$ converges to some limit probability distribution (denote it Q^*) and when $k \to \infty$, we obtain:

$$Q_{(k)}(X_R | CT) \to Q_{(k-1)}(X_R | CT) \qquad (13)$$

Finally, we show that this Q^* fulfills all given constraints, using (13) together with (7), we have:

$$\beta_{k-1}(\pi_{V_i}) \cdot \frac{R(V_i | L_i)}{Q_{(k-1)}(V_i | L_i, CT)} \to 1 \qquad (14)$$

When $k \to \infty$, we also have $Q_{(k)}(CT) \to Q_{(k-1)}(CT)$, so:

$$\beta_{k-1}(\pi_{V_i}) \to \alpha_{k-1}(\pi_{V_i}) \qquad (15)$$

From (14) and (15), we have:

$$\alpha_{k-1}(\pi_{V_i}) \cdot \frac{R(V_i \mid L_i)}{Q_{(k-1)}(V_i \mid L_i, CT)} \to 1 \text{ , i.e.}$$

$$Q_{(k-1)}(V_i \mid L_i, CT) \to \alpha_{k-1}(\pi_{V_i}) \cdot R(V_i \mid L_i)$$

Since both $Q_{(k-1)}$ and $R(V_i \mid L_i)$ are probability distributions, then the normalization factor $\alpha_{k-1}(\pi_{V_i}) \to 1$, then we have:

$$\lim_{k\to\infty} Q_{(k)}(V_i \mid L_i, CT) = R(V_i \mid L_i)$$

□

E. An Example

We demonstrate the validity of our approach by a simple example ontology. In this ontology, "Animal" is a primitive concept class; "Male", "Female", "Human" are subclasses of "Animal"; "Male" and "Female" are disjoint with each other; "Man" is the intersection of "Male" and "Human"; "Woman" is the intersection of "Female" and "Human"; "Human" is the union of "Man" and "Woman".

The following constraints or probabilities are attached to X_R = {Animal, Male, Female, Human, Man, Woman}:

(1) P(Animal) = 0.5;
(2) P(Male|Animal) = 0.5;
(3) P(Female|Animal) = 0.48;
(4) P(Human|Animal) = 0.1;
(5) P(Man|Human) = 0.49;
(6) P(Woman|Human) = 0.51.

We obtained the BN by first constructing the DAG (as described by Section III), then the CPT for nodes in X_C (as described in Subsection IV.A), and finally approximating the CPTs of nodes in X_R by running D-IPFP. Fig.5 below shows the BN we obtained. It can be seen that, when all control nodes are set to True, the conditional probability of "Male", "Female", and "Human", given "Animal", are 0.5, 0.48, and 0.1, respectively, the same as the given probability constraints. All other constraints, which are not shown in the figure due to space limitation, are also satisfied.

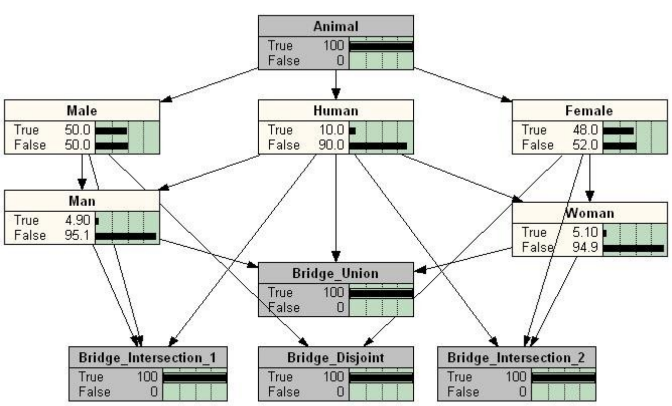

Fig.5. – An Example

The initial CPTs (of nodes in X_R) used in this example and the final solution CPTs (of nodes in X_R) obtained by D-IPFP are listed in Table 6. Note that in all initial CPT, values on the first row were set to 0.5. They can be set to any arbitrary values greater than 0 and less than 1. Values for all other rows were set according to the subclass relation. It can be seen that the values on the first row in all CPT have been changed from their initial values.

Table 6 – CPT of the Example Ontology

Initial arbitrary CPT

Animal	
True	False
0.5	0.5

Animal	Male True	Male False
True	0.5	0.5
False	0	1

Animal	Female True	Female False
True	0.5	0.5
False	0	1

Animal	Human True	Human False
True	0.5	0.5
False	0	1

Male	Human	Man True	Man False
True	True	0.5	0.5
True	False	0	1
False	True	0	1
False	False	0	1

Female	Human	Woman True	Woman False
True	True	0.5	0.5
True	False	0	1
False	True	0	1
False	False	0	1

Final CPT obtained by D-IPFP

Animal	
True	False
0.92752	0.07248

Animal	Male True	Male False
True	0.95677	0.04323
False	0	1

Animal	Female True	Female False
True	0.95469	0.04531
False	0	1

Animal	Human True	Human False
True	0.18773	0.81227
False	0	1

Male	Human	Man True	Man False
True	True	0.47049	0.52951
True	False	0	1
False	True	0	1
False	False	0	1

Female	Human	Woman True	Woman False
True	True	0.51433	0.48567
True	False	0	1
False	True	0	1
False	False	0	1

F. Discussion over D-IPFP

Some other general optimization methods such as simulated annealing (SA) and genetic algorithm (GA) can also be used to construct CPTs of the regular nodes in the translated BN. However, they are much more expensive and the quality of results is often not guaranteed. In our experiments, D-IPFP converges quickly (in seconds, most of the time in less than 30 iterative steps), despite its exponential time complexity in theoretical analysis. The space complexity of D-IPFP is trivial since each time we only manipulate the CPT of one node, not the entire joint probability table.

However some theoretical issues regarding D-IPFP remain to be addressed, including the existence and uniqueness of the solution and the impact of the input constraint set on the quality of the solution:

(1) Existence: Under what condition will the input constraint set specify a multivariate joint distribution?
(2) Uniqueness: Assume such joint distribution exists, will it be unique?
(3) Quality of input set: How to deal with weakly consistent, inconsistent or incomplete input set?

Future work also includes extending D-IPFP to handle an input set with constraints of more general form, such as: $\{P(A \mid B)\}$, where $A, B \subseteq X_R = \{V_1, ..., V_s\}$, $A \cap B = \varnothing$. This might be possible since according to the chain rule, $P(V_{i_1}, ..., V_{i_l} \mid B)$ can be transformed into a set of constraints with the form of $P(V_i \mid C)$, $C \subseteq \{V_1, ..., V_s\} \setminus \{V_i\}$, i.e.

$P(V_{i_1} \mid B, V_{i_2}, ..., V_{i_I})$

$P(V_{i_2} \mid B, V_{i_3}, ..., V_{i_I})$

$\ldots$

$P(V_{i_{I-1}} \mid B, V_{i_I})$

$P(V_{i_I} \mid B)$

In our experiments, we also notice that the order to apply the constraints will not affect the solution, and the values of the initial distribution $Q_{(0)}(X) = P_{init}(X)$ (but avoid 0 and 1) will not affect the solution either.

V. Reasoning

The probabilistic-extended ontology can supports common ontology-related reasoning tasks in the subspace of CT. Here we outline how three such tasks can be done in principle. Detailed algorithms are under development.

A. Concept Satisfiability

Given a concept represented by a description e, decide whether $P(e \mid CT) = 0$ (False). $P(e \mid CT)$ can be computed by applying the chain rule of BN.

B. Concept Overlapping

The degree of the overlap or inclusion between a concept C and a description e can be measured by $P(c \mid e, CT)$, which can be computed by applying general BN belief update algorithms (c means the "True" state of C).

C. Concept Subsumption

Find the most similar concept C that a given description e belongs to. This task cannot be done by simply computing the posterior probability $P(C \mid e, CT)$, because any class node would have higher probability (prior or posterior) than its children, and the root node always has the probability of 1. Instead, we define a similarity measure $MSC(e, C)$ between e and C based on *Jaccard Coefficient* [16]:

$$MSC(e,C) = P(e \cap C \mid CT) / P(e \cup C \mid CT)$$

$$= P(e,c \mid CT) / (P(e \mid CT) + P(c \mid CT) - P(e,c \mid CT)) \qquad (16)$$

This measure is an intuitive and easy-to-compute measure, and when e is a subclass of C (i.e., $P(c \mid e, CT) = 1$), it reduces to the Most-Specific-Subsumer of DL. Otherwise, C is a class that has the largest overlap with e. We are also looking at other similarity measures, such as those based on entropy or mutual information.

In our example ontology (see Fig.5), to find the concept that is most similar to the given description $e = \neg Man \cap Animal$, we compute the similarity measure of e and each of the nodes in X_R = {Animal, Male, Female, Human, Man, Woman} using (16):

$MSC(e, Animal) = 0.4755$,

$MSC(e, Male) = 0.4506$,

$MSC(e, Female) = 0.5047$,

$MSC(e, Human) = 0.0510$,

$MSC(e, Man) = 0.0$,

$MSC(e, Woman) = 0.0536$.

This leads us to conclude that class "Female" is the most similar concept to e, since it has the highest similarity measure among all nodes in this particular example.

VI. Conclusion, Related Work and Discussion

In this paper we present our ongoing research on probabilistic extension to OWL. We have defined new OWL classes ("PriorProb", "CondProb", and "Variable"), which can be used to markup probabilities for classes in OWL files. We have also defined a set of rules for translating OWL ontology taxonomy into Bayesian network DAG and provided a new algorithm D-IPFP to construct CPTs for all the regular nodes.

Our probabilistic extension to OWL is compatible with OWL semantics, and the translated BN is associated with a joint probability distribution over the application domain consistent with given probabilities. We are currently actively working on extending the translation to include properties, developing algorithms to support common ontology-related reasoning tasks, and formalizing mapping between two ontologies as probabilistic reasoning across two translated BN. Based on successful resolution of these issues and other refinement of our framework, we plan to implement a prototype which can automatically translate a given OWL ontology with uncertainty information into a BN and can also support common ontology-based reasoning tasks.

Researchers in the past have attempted to apply different formalisms such as fuzzy logic, rough set theory, and Bayesian probability as well as ad hoc heuristics into ontology definition and reasoning (see [10] for a brief survey). Works that integrate probabilities into description logic based systems (e.g., [9, 11, 12, 13, 14] are particularly relevant to our work. Works in [12, 13] provide a probabilistic extension of the DL *ALC* based on probabilistic logics. P-CLASSIC [14] gives an informal probabilistic extension to CLASSIC also based on Bayesian networks, in which each probabilistic component is associated with a set of p-classes, each of which is represented using a BN. P-*SHOQ*(D) [11] is the probabilistic extension of DL *SHOQ*(D) [15] based on the notion of probabilistic lexicographic entailment from probabilistic default reasoning. Among these works, only P-*SHOQ*(D) is able to represent assertional (i.e., Abox) probabilistic knowledge about concept and role instances. The primary difference between [9] and our work is that their links are pointed from subconcepts to superconcepts, which makes the construction of CPTs difficult. Our method are not aimed at providing additional means to represent uncertainty or probabilistic aspect of the domain but rather at developing formal rules to directly translate an OWL ontology into a Bayesian network.

Acknowledgment

This work was supported in part by DARPA contract

F30602-97-1-0215. The software system we used to construct Bayesian networks in our examples is Netica from Norsys Software Corporation (http://www.norsys.com/). We would like to thank Dr. Hans-Hermann Bock and Dr. Erhard Cramer for their comments about IPFP.

REFERENCES

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

[2] Z. Ding and Y. Peng, "A Probabilistic Extension to Ontology Language OWL", in *Proceedings of the 37th Hawaii International Conference on System Sciences*. Jan. 2004, Big Island, HI.

[3] R. Kruithof, "Telefoonverkeersrekening", *De Ingenieur* 52, E15-E25, 1937.

[4] W. E. Deming and F. F. Stephan, "On a Least Square Adjustment of a Sampled Frequency Table when the Expected Marginal Totals are Known", *Ann. Math. Statist.* 11, pp. 427-444, 1940.

[5] I. Csiszar, "*I*-divergence Geometry of Probability Distributions and Minimization Problems", *The Annuals of Probability*, vol. 3, no.1, pp. 146-158, Feb. 1975.

[6] J. Vomlel, "Methods of Probabilistic Knowledge Integration", PhD Thesis, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, December 1999.

[7] H.-H. Bock, "A Conditional Iterative Proportional Fitting (CIPF) Algorithm with Applications in the Statistical Analysis of Discrete Spatial Data", *Bull. ISI*, *Contributed papers of 47th Session in Paris*, vol. 1, pp. 141-142, 1989.

[8] E. Cramer, "Probability Measures with Given Marginals and Conditionals: *I*-projections and Conditional Iterative Proportional Fitting", *Statistics and Decisions*, vol. 18, pp. 311-329, 2000.

[9] M. Holi and E. Hyvönen, "Probabilistic Information Retrieval Based on Conceptual Overlap in Semantic Web Ontologies", in *Proceedings of the 11th Finnish AI Conference*, *Web Intelligence*, vol. 2, Finnish AI Society, Finland, 2004.

[10] H. Stuckenschmidt and U. Visser, "Semantic Translation based on Approximate Re-classification", in *Proceedings of the Workshop "Semantic Approximation, Granularity and Vagueness", KR'00*.

[11] R. Giugno and T. Lukasiewicz, "P-*SHOQ*(D): A Probabilistic Extension of *SHOQ*(D) for Probabilistic Ontologies in the Semantic Web", INFSYS Research Report 1843-02-06, Wien, Austria, April 2002.

[12] J. Heinsohn, "Probabilistic Description Logics", in *Proceedings of UAI-94*, pp. 311-318, 1994.

[13] M. Jaeger, "Probabilistic Reasoning in Terminological Logics", in *Proceedings of KR-94*, pp. 305-316, 1994.

[14] D. Koller, A. Levy and A. Pfeffer, "P-CLASSIC: A Tractable Probabilistic Description Logic", in *Proceedings of AAAI-97*, pp. 390-397, 1997.

[15] I. Horrocks and U. Sattler, "Ontology Reasoning in the *SHOQ*(D) Description Logic", in *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001.

[16] C. J. van Rijsbergen, *Information Retrieval*. Lodon:Butterworths, Second Edition, 1979.

[17] Semantic Web Homepage: http://www.w3.org/2001/sw/

[18] W3C WebOnt Working Group: http://www.w3.org/2001/sw/WebOnt/

[19] W3C RDF Homepage: http://www.w3.org/RDF/