WUJNS
Wuhan University Journal of Natural Sciences

Vol. 11 No. 5 2006 1132-1136

Article ID: 1007-1202(2006)05-1132-05

Uncertainty Modeling Based on Bayesian Network in Ontology Mapping

□ LI Yuhua, LIU Tao, SUN Xiaolin
College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China

Abstract: How to deal with uncertainty is crucial in exact concept mapping between ontologies. This paper presents a new framework on modeling uncertainty in ontologies based on bayesian networks (BN). In our approach, ontology Web language (OWL) is extended to add probabilistic markups for attaching probability information, the source and target ontologies (expressed by patulous OWL) are translated into bayesian networks (BNs), the mapping between the two ontologies can be digged out by constructing the conditional probability tables (CPTs) of the BN using a improved algorithm named I-IPFP based on iterative proportional fitting procedure (IPFP). The basic idea of this framework and algorithm are validated by positive results from computer experiments.

Key words: uncertainty; Bayesian network; conditional probability table (CPT); improved-iterative proportional fitting procedure (I-IPFP)

CLC number: TP 301.6

Received date: 2006-03-15
Foundation item: Supported by the National Natural Science Foundation of China (60403027)

Biography: LI Yuhua(1968-), female, Associate professor, research direction: semantic Web, data mining. E-mail: yuhua_yy@163.com

0 Introduction

In many semantic interoperability applications, ontology mapping is the first step to be resolved[1]. If we want to get exact mapping information, we need to deal with the problem of uncertainty[2].

Uncertainty becomes more prevalent in concept mapping between two ontologies. Semantic similarities between concepts are difficult to represent logically, but can easily be represented probabilistically. This has motivated recent development of ontology mapping taking probabilistic approaches, such as Gay and Lesbian University Employees (GLUE) and Ontology Mapping Enhancer (OMEN)[3-5]. However, these existing approaches fail to completely address uncertainty in mapping.

The work reported in this paper involved in a number of significant ways, in which uncertainty in ontology mapping can be dealt with properly. Our system framework consists of three components: ①an ontology encoding module to change the raw ontology to a ontology with probability; ②a transition part to translate given ontologies into Bayesian networks (BNs); ③a concept mapping module that takes a set of raw similarities learned from domain knowledge or given by experts as input and then finds mappings between concepts from two different ontologies based on evidential reasons across two BNs can be found.

1 Technology Background

1.1 Web Ontology Language

Web Ontology Language (OWL) is designed to be utilized by users who need to process the content of information instead of just presenting information to humans. OWL facilitates have greater machine interpretability of Web content than that supported by Extensible Markup Language (XML), Resource Description Framework (RDF) and Resource Description Framework Schema (RDFS) by providing additional vocabulary along with a formal semantics[6].

1.2 Bayesian Network

Generally, a Bayesian Network (BN) of n variables consists of a Directed Acyclic Graph (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 variable to the other[7]. The uncertainty of the relationship is represented by the conditional probability table (CPT) $P(X_i \mid T_i)$ associated with each node X_i, where T_i is the parent set of X_i. Under a conditional independence assumption, the joint probability distribution of $X = \{X_1, \cdots, X_n\}$ can be factored out as a product of the CPTs: $P(X = x) = \prod_{i=1}^{n} P(x_i \mid T_i)$.

1.3 Iterative Proportional Fitting Procedure (IPFP)

For a given distribution $Q_0(x)$ and consistent constraints R, IPFP converges to $Q^*(x)$ that is a projection of Q_0 on R. This is done by iteratively modifying the distributions according to the following equation, each time using one constraint in R:

$$Q_k(x) = Q_{k-1}(x) \cdot \frac{R_i(y)}{Q_{k-1}(y)} \qquad (1)$$

Where m is the number of constraints R, and $i = ((k-1) \bmod m) + 1$ determines the constraint used at step k[8].

2 Encoding Probabilities in OWL

In our approach, OWL is extended to augment probability information. These probabilities can be either provided by domain experts or learned from Web data as described in the previous section.

For a concept class C and its parent concept class set S_C, two probabilities are as follows:

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

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

To add such uncertainty information into an existing ontology, we should treat probability as a kind of resource, two OWL classes ("PriorProb", "CondProb") are augmented[9].

A probability with the form $P(C)$ is defined as an instance of class "PriorProb", which has two mandatory properties: "hasVarible" and "hasProbValue".

For example, $P(C) = 0.8$, the prior probability, which is 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〉

A probability with such a form is defined as an instance of class "CondProb", which has three properties: "hasCondition", "hasVariable" and "hasProbValue". The range of properties "hasCondition" and "hasVariable" is a defined class named "Variable", which has two properties: "hasClass" and "hasState". "hasClass" points to the concept class about this probability and "hasState" gives the "True" (belong to) or "False" (not belong to) state of this probability.

3 System Framework

3.1 Encoding and Pre-Processing

In our framework, the resource and target ontology should be encoded into a new ontology with probability information. The information can be obtained by learning from Web ontology information or being defined by experts. After this encoding module, the ontology with probability has to be checked through syntax checker and semantic checker, then can be translated to BNs.

3.2 Structural Translation

A set of translation rules is developed to convert an OWL ontology into a DAG of BN.

The general principle underlying these rules is that all classes are translated into nodes in BN, and an arc is drawn between two nodes in BN, if the two corresponding classes are related by a "predicate" in the OWL file[10], with the direction from the superclass to the subclass. Control nodes are created during the translation to facilitate modeling relations among class nodes that are specified by OWL logical operators, and there is a converging connection from each concept nodes involved in this logical relation to its specific control node. There are five types of control nodes in total, which correspond to the five types of logical relations: They are: "and" (owl: intersectionOf), "or" (owl: unionOf), "not" (owl: complementOf), "disjoint" (owl: disjointWith) and "same as" (owl: equivalentClass).

3.3 Constructing Conditional Probability Tables

The nodes in the DAG obtained from the structural translation step can be divided into two disjoint groups: X_R, nodes representing concepts in ontology, and X_C, control nodes for bridging logical relations. The CPT for a control node in X_C can be determined by the logical relation it represents so that when its state is "True", the corresponding logical relation holds among its parent nodes. When all the control nodes' states are set to "True" (denote this state as CT), all the logical relations defined in the original ontology are held in the translated BN[11]. The remaining issue is then to construct the CPTs for each node in X_R so that $P(X_R | \mathrm{CT})$, the joint distribution of all regular nodes in the subspace of CT.

Based on this structural translation rules, there are five types of control nodes corresponding to the five logic operators in OWL. They are "Complement", "Disjoint", "Equivalent", "Intersection" and "Union". Their CPTs are determined by the logical relation among its parent concept class nodes, which are to be specified later.

Figure 1 below is a BN translated from a simple 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 probability constraints are attached to:

$X_R = \{\text{Animal}, \text{Male}, \text{Female}, \text{Human}, \text{Man}, \text{Woman}\}$

$X_C = \{\text{Disjoint}, \text{Intersection}, \text{Union}\}$

$P(\text{Animal}) = 0.50$; $P(\text{Male} | \text{Animal}) = 0.50$;

$P(\text{Female} | \text{Animal}) = 0.48$; $P(\text{Human} | \text{Animal}) = 0.10$;

$P(\text{Man} | \text{Human}) = 0.49$; $P(\text{Woman} | \text{Human}) = 0.51$.

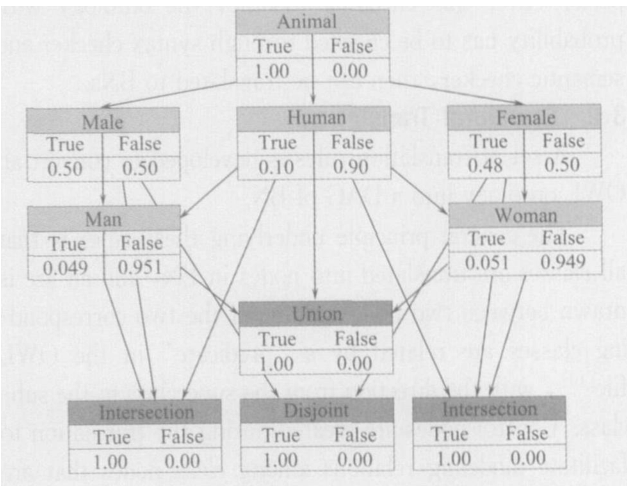

Fig. 1 A translation example

3.4 Improved-Iterative Proportional Fitting Procedure

The issue is to construct CPTs for the regular nodes in X_R so that $P(X_R | \mathrm{CT})$, 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. To address these issues, we developed an algorithm (I-IPFP) to approximate these CPTs for X_R based on the IPFP.

First we divide constraints into two types. $R_i(y)$ is said to be local if Y contains nothing else except one variable X_j and zero or more of its parents. Otherwise, $R_i(y)$ is said to be non-local. How to deal with local and non-local constraints in I-IPFP is given in the next two subsections.

①Local constraints

Suppose $Q_{k-1} = \prod_{i=1}^{n} Q_{k-1}(x_i | T_i)$. Consider a local constraint $R_i(y) = R_i(x_j, z^j \subseteq T_j)$. Since it is a constraint only on x_j and some of its parents, updating $Q_{k-1}(x)$ by $R_i(y)$ can be done by only updating $Q_{k-1}(x_j | T_j)$, the CPT for x_j, while leaving all other CPTs intact.

Since $Q_{k-1}(x_j | T_j)$ is an conditional distribution on x_j, $Q_{k-1}(x_j | T_j) R_i(y) / Q_{k-1}(y)$ is in general not a probability distribution, and thus cannot be used as the CPT for X_j in $Q_k(x)$. This problem can be resolved by normalization. The update rule becomes:

$$\begin{cases} Q_k(y | s) = Q_{k-1}(y | s) \cdot \dfrac{R_i(y)}{Q_{k-1}(y)} \cdot \alpha_k \\ Q_k(x_l | T_l) = Q_{k-1}(x_l | T_l) \ \forall x_l \notin y \end{cases} \quad (2)$$

where

$$\alpha_k = \sum_{x_j} Q_{k-1}(x_j | T_j) \cdot \frac{R_i(y)}{Q_{k-1}(y)} \quad (3)$$

Since only the CPT for X_j is changed, this rule leads to

$$Q_k(x) = Q_k(x_j | T_j) \cdot \prod_{l \neq j} Q_{k-1}(x_l | T_l) \quad (4)$$

Therefore $Q_k(x)$ is consistent with G_0, it satisfies the structural constraint.

②Non-local constraints

Now we generalize the idea of rule (2) to non-local constraints. Without loss of generality, consider one such constraint $R_i(y)$ where Y spans more than one CPT. Let multiply all CPTs for variables in Y, one can construct a conditional distribution

$$Q'_{k-1}(Y \mid S) = \prod_{X_j \in Y} Q_{k-1}(x_j \mid T_j) \quad (5)$$

With equation (5), we define

$$Q'_{k-1}(x) = Q_{k-1}(x) = Q'_{k-1}(y \mid s) \cdot \prod_{X_l \notin Y} Q_{k-1}(x_l \mid T_l) \quad (6)$$

Now $R_i(y)$ becomes local to the table $Q'_{k-1}(y|s)$, we can obtain $Q'_k(x)$ by obtaining $Q'_k(y|s)$ using the Eq. (2) for local constraint.

$$\begin{cases} Q'_k(y \mid s) = Q'_{k-1}(y \mid s) \cdot \dfrac{R_i(y)}{Q'_{k-1}(y)} \cdot \alpha_k \\ Q'_k(x_l \mid T_l) = Q'_{k-1}(x_l \mid T_l) \ \forall x_l \notin y \end{cases} \quad (7)$$

Next, we extract $Q_k(x_j|T_j)$ for all $X_j \in Y$ from $Q'_k(y|s)$ by $Q_k(x_j|T_j) = Q'_k(x_j|T_j)$.

The process ends with:

$$Q_k(x) = \prod_{X_j \in Y} Q'_k(x_j \mid T_j) \cdot \prod_{X_l \notin Y} Q_{k-1}(x_l \mid T_l) \quad (8)$$

Update of $Q_{k-1}(x)$ to $Q_k(x)$ by $R_i(y)$ can be seen to consist of three steps:

a) get $Q'_{k-1}(y|s)$ from CPTs for $X_j \in Y$ by Eq. (5);

b) update $Q'_{k-1}(y|s)$ to $Q'_k(y|s)$ by $R_i(y)$ using Eq. (7);

c) extract $Q'_k(x_j|T_j)$ from $Q'_k(y|s)$ by Eq. (8).

Comparing Eqs. (5), (7) and (8), this procedure of I-IPFP amounts to an iteration of a local IPFP on $Q'_{k-1}(y|s)$.

③Algorithm I-IPFP

I-IPFP $(N_0(X), R=\{R_1, R_2, \cdots, R_m\})\{$

Step 1 $\quad Q_0(x) = \prod_{i=1}^{n} Q_0(x_i \mid T_i)$

Step 2 $\{$

$i=((k-1) \bmod m)+1;$

If $R_i(y=(x_j, z^j \subseteq T_j))\ \{$

$$Q_k(x_j \mid T_j) = Q_{k-1}(x_j \mid T_j) \cdot \frac{R_i(y)}{Q_{k-1}(y)} \cdot \alpha_k;$$

$$Q_k(x_l \mid T_l) = Q_{k-1}(x_l \mid T_l) \ \forall l \neq j;$$

$\}$

$\{$

$$Q'_{k-1}(y \mid s) = \prod_{X_j \in Y} Q_{k-1}(x_j \mid T_j);$$

$$Q'_k(y \mid s) = Q'_{k-1}(y \mid s) \cdot \frac{R_i(y)}{Q_{k-1}(y)} \cdot \alpha_k;$$

$$Q_k(x_j \mid T_j) = Q'_k(x_j \mid T_j) \ \forall x_j \in y;$$

$$Q_k(x_l \mid T_l) = Q_{k-1}(x_l \mid T_l) \ \forall x_l \in y;$$

$$Q_{k-1}(x_j \mid T_j) = Q_k(x_j \mid T_j) \ \forall x_j;$$

$\}$

$k++;$

$\}$

Step 3 Return $N^*(X)$.

$\}$

4 Experiment

4.1 Analysis of Algorithm Efficiency

①IPFP

The computation of IPFP is on the entire joint distribution of X at every iteration. Roughly speaking, when $Q_{k-1}(x)$ is modified by constraint $R_i(y)$, Eq. (1) requires to check each entry in $Q_{k-1}(x)$ against every entry of $R_i(y)$ and make the update if x is consistent with y. The cost can be estimated as $O(2^n \times 2^{|Y|})$.

②I-IPFP

The moderate sacrifice for I-IPFP is rewarded by a significant saving in computation. Since $R_i(y)$ is now used to modify $Q'_{k-1}(y|s)$, not $Q_{k-1}(x)$, the cost for each step is reduced from $O(2^n \cdot 2^{|Y|})$ to $O(2^{|s|+|y|} \cdot 2^{|y|})$ where $O(2^{|s|+|y|})$ is the size of CPT $Q'_{k-1}(y|s)$. The saving is $O(2^{n-|s|+|y|})$.

4.2 Comparison of IPFP and I-IPFP

We choose different numbers of the BN structure's nodes, and record the executive time by the different algorithm IPFP and I-IPFP. The experiment's result is given in Fig. 2.

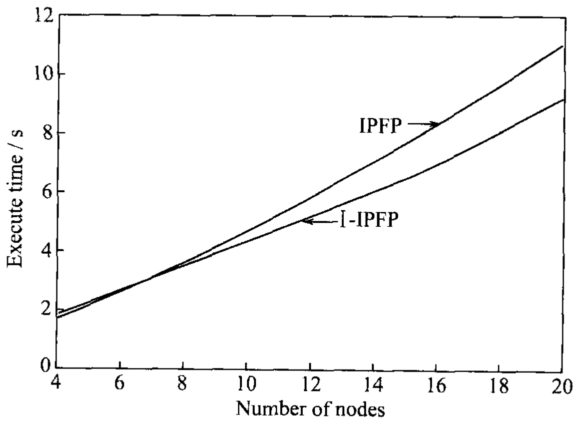

Fig. 2 Comparison of execute time

The experiment's result shows that the efficiency of IPFP precedes that of I-IPFP when the number of nodes is small, on the contrary the efficiency of I-IPFP excels that of IPFP when the number of nodes exceeds the critical value, and the larger the number is, the more effective I-IPFP is.

5 Conclusion

In this paper we present research on probabilistic extension to OWL. We have defined new OWL classes that 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 DAG and provided a new algorithm I-IPFP to construct CPTs for all the regular nodes. The translated BN is associated with a joint probability distribution over the application domain consistent with given probabilities. Finally we validate our method by doing experiments, and give a comparison of the algorithm IPFP and improved one I-IPFP.

In the future we are going to work on improving efficiency of the algorithm continually to satisfy the increasing number of nodes.

References

[1] Noy N. Semantic Integration: A Survey of Ontology-Based Approaches [J]. *SIGMOD Record*, 2004, **33**(4): 65-70.

[2] Ding Z, Peng Y. A Probabilistic Extension to OWL [C]// *Proceedings of the 37th Hawaii International Conference on System Sciences*. Big Island, Hawaii, USA, Jan. 2004.

[3] Pan R. A Framework for Bayesian Network Mapping [C]// *Proceedings of 2005 American Association for Artificial Intelligence*. Pittsburgh, Pennsylvania, USA, April 2005.

[4] Holi M, Hyvönen E. Probability Measures with Given Marginals and Conditionals: I-Projections and Conditional Iterative Proportional Fitting [J]. *Statistics and Decisions*, 2000, **18**(2): 311-329.

[5] Csiszar I. I-Divergence Geometry of Probability Distributions and Minimization Problems[J]. *The Annuals of Probability*, 1975, **3**(1): 146-158.

[6] Koller D, Levy A. P-CLASSIC: A Tractable Probabilistic Description Logic [C]// *Proceedings of 1997 American Association for Artificial Intelligence*. Providence, Rhode Island, USA, Aug. 1997.

[7] Lacher M, Groh G. Facilitating the Exchange of Explicit Knowledge through Ontology Mappings [C]// *Proceedings of the 14th International Florida Artificial Intelligence Research Society Conference*. Melbourne Beach, Florida, USA, May 2001.

[8] Mitra P, Noy N. OMEN: A Probabilistic Ontology Mapping Tool [C]// *Workshop on Meaning Coordination and Negotiation at the Third International Conference on the Semantic Web*. Hisroshima, Japan, Oct. 2003.

[9] Holi M, Hyvönen E. Probabilistic Information Retrieval Based on Conceptual Overlap in Semantic Web Ontologies [C]// *Proceedings of the 11th Finnish Artificial Intelligence Conference*. Vantaa, Finland, Sept. 2004.

[10] Prasad S, Peng Y, Finin T. A Tool for Mapping between Two Ontologies Using Explicit Information [C]// *Autonomous Agents and Multi-Agent Systems 2002 Workshop on Ontologies and Agent Systems*. Bologna, Italy, July 2002.

[11] Melnik S, Garcia-Molina H, Rahm E. Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching [C]// *18th International Conference on Data Engineering*. San Jose, USA, April 2005.

□