Proceedings of the 2007 Industrial Engineering Research Conference
G. Bayraksan, W. Lin, Y. Son, and R. Wysk, eds.

A Probabilistic Framework for Semantic Similarity and Ontology Mapping

Yun Peng, Zhongli Ding, Rong Pan, Yang Yu
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250, USA

Boonserm Kulvatunyou, Nenad Ivezic, Albert Jones
Manufacturing Systems Integration Division
National Institute of Standards and Technology (NIST)
MS 8265, Gaithersburg, MD 20899, USA

Hyunbo Cho
Department of Industrial and Management Engineering
Pohang University of Science and Technology,
Pohang, South Korea

Abstract

We propose a probabilistic framework to address uncertainty in ontology-based semantic integration and interoperation. This framework consists of three main components: 1) *BayesOWL* that translates an OWL ontology to a Bayesian network, 2) *SLBN* (Semantically Linked Bayesian Networks) that support reasoning across translated BNs, and 3) a *Learner* that learns from the web the probabilities needed by the other two components. This framework expands the semantic web and can serve as a theoretical basis for solving real world semantic integration problems.

Keywords
Semantic web, uncertainty, integration, ontology, Bayesian networks

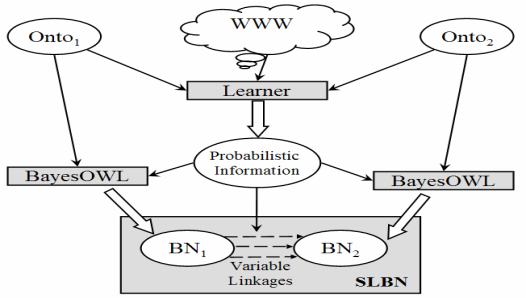
1. Uncertainty in Ontology Mapping and Semantic Integration

Representing and reasoning with uncertainty have been realized as an important issue in a single ontology [4, 11]. For example, in *ontology construction*, besides knowing that "*A* is a subclass of *B*" one may also know and wish to express in the ontology how likely an instance of *B* belongs to *A*. In *ontology reasoning*, one may want to infer not only if *A* subsumes *B*, but also the degree of closeness of *A* to *B*, or one may want to know the degree of similarity between *A* and *B* even if *A* and *B* are not subsumed by each other. Uncertainty becomes more prevalent in *concept mapping* between two ontologies. In many applications, exact matches between concepts defined in two ontologies do not exist. Instead, a concept defined in one ontology may find *partial* matches to one or more concepts in another ontology, often with different degrees of similarity.

How to provide consistent and unified semantic support for information and knowledge integration that handles uncertainty in a principled and practical manner is the problem our research attempts to address. The approach we take is probabilistic, and Bayesian networks (BN) are taken as the formalism for modeling the probabilistic interdependencies among ontological entities. This paper presents the probabilistic framework developed in this research effort.

2. Overview of Our Probabilistic Framework

We assume the ontologies are written in OWL (Web Ontology Language, http://www.w3.org/TR/owl-features/). Fig. 1 below gives an overview of this framework in the context of ontology mapping. The three main components, *BayesOWL*, *SLBN* (semantically linked BN), and the *Learner*, are described in detail in the next three sections.



- *BayesOWL* translates two ontologies Onto1 and Onto2 into BN1 and BN2;
- *SLBN* supports concept mapping between Onto1 and Onto2 as probabilistic reasoning between BN1 and BN2;
- *Learner* learns probabilities for BayesOWL and SLBN from text exemplars searched from the web.

Figure 1: Overview of the probabilistic framework

3. BayesOWL

To translate an OWL ontology to a BN, BayesOWL [2] takes two inputs: 1) the OWL file that defines the ontology, and 2) a collection of probabilistic constraints, including prior probabilities of concept classes and conditional probabilities of superclass relations defined in the ontology. A set of *structural translation rules* are called to build the BN structure (a directed acyclic graph or DAG) from the ontology definition. Conditional probability tables (CPTs) of the BN are then constructed based on the DAG and the probabilistic constraints.

Probability information markups. We represent the semantics of probabilistic constraints as follows. We treat concept classes *A* and *B* in an ontology as random binary variables and interpret $P(A=a)$ (or $P(A=\neg a)$) as the prior probability that an arbitrary individual belongs (or does not belong) to class *A*, and $P(a \mid b)$ as the conditional probability that an individual of class *B* also belongs to class *A*. These two types of probabilities for classes and superclass relations in an ontology are most likely to be available to ontology designers. To add such uncertainty information into an existing ontology, we treat a probability as a kind of resource, and define two OWL classes *PriorProb* and *CondProb* for their encoding. Class *PriorProb* has two mandatory properties: *hasVarible* and *hasProbValue*, while class *CondProb* has three mandatory properties: *hasVariable*, *hasCondition*, and *hasProbValue*. For example, $P(c)=0.8$ for class C can be expressed as

```
<Variable rdf:ID="c">
   <hasClass>C</hasClass>
   <hasState>True</hasState>
</Variable>
```

```
<PriorProb rdf:ID="P(c)">
   <hasVariable>c</hasVariable>
   <hasProbValue>0.8</hasProbValue>
</PriorProb>
```

Conditional probabilities can be encoded in a similar fashion. (See [2] for more details on probability markups.)

Bayesian network (BN) is a graphic model for probabilistic interdependencies among a set of random variables [9]. A BN consists of two parts: 1) a directed acyclic graph (DAG) in which nodes represent variables and directed arcs between nodes signify the dependencies; and 2) a conditional probability table (CPT) $P(x_i \mid p_i)$ for each variable x_i, given all its parent nodes p_i. Based on an independent assumption, the joint distribution of all variables can be computed from the local CPTs by the chain rule $P(X=x)=\prod_{i=1}^{n} P(x_i \mid p_i)$. Constructions of DAG and CPT are given next.

Structural translation. The ontology augmented with probability constraints is still an OWL file. It can be translated into a BN by first forming a DAG following a set of rules. Special nodes, call *L-Nodes*, are created during the translation to facilitate modeling relations among class nodes that are specified by OWL *logical* operators (union, intersection, complement, disjoint, equivalent). These structural translation rules are summarized as follows.

(1) Every concept class *C* is mapped into a binary variable node in the translated BN.
(2) Constructor *rdfs:subClassOf* is modeled by an arc from the superclass node to the subclass node.
(3) A concept class *C* defined as the intersection of concept classes C_i ($i = 1,\ldots,n$) is mapped into a subnet in the translated BN with one arc from each C_i to *C*, and one arc from *C* and each C_i to an L-Node called *LNodeIntersection*. Constructor *owl:UnionOf* is modeled in the same way except now the directions of arcs between *C* and each C_i are reversed.
(4) If two concept classes *C*1 and *C*2 are related by constructors *owl:complementOf*, *owl:equivalentClass*, or *owl:disjointWith*, then an L-Node (named *LNodeComplement, LNodeEquivalent,* and *LNodeDisjoint*, respectively) is added to the translated BN with directed links from *C*1 and *C*2 to the L-Node.

Constructing CPT. The nodes in the DAG generated from the structural translation can be divided into two disjoint groups: X_C for nodes representing concepts in the ontology, and X_L for L-Nodes for logical relations. The CPT for an L-Node in X_L can be determined by the logical relation it represents; in other words, the entries in $P(x_i \mid \pi_i)$ shall be filled in such a way that when the state of x_i is "True" the intended logical relation holds among its parent nodes. When all L-Nodes are set to "True" (denoting this situation as *LT*), all the logical relations defined in the original ontology are held in the translated BN. Constructing the CPT $P(x_i \mid \pi_i)$ for a concept node $x_i \in X_C$ is more complicated. It must satisfy the given probabilistic constraints of the prior $P(x_i)$ and conditionals $P(x_i \mid x_j)$ for all $x_j \in \pi_i$; and this has to be done in the subspace of *LT*. In other words, we now have a multi-constraint satisfaction problem: construct $P(x_i \mid \pi_i)$ for all $x_i \in X_C$ such that $P(X_C \mid LT)$ is consistent with all given probabilistic constraints.

We apply the technique known as Iterative Proportional Fitting Procedure (IPFP) [2, 5] to construct CPTs for concept nodes in X_C. IPFP is a procedure that modifies a given probability distribution (PD) $P(X)$ to satisfy a set of constraints $\boldsymbol{R}=\{R(Y_i)\}$, each of which is a prior or conditional distribution on a subset of variables $Y_i \subseteq X$. Briefly, IPFP starts with the initial PD $Q_0(x) = P(X)$, and at each iteration it modifies the PD to satisfy one constraint $R(Y_i)$ by

$$Q_k(X) = Q_{k-1}(X) \cdot R(Y_i) / Q_{k-1}(Y_i) \tag{1}$$

It can be shown that a consistent set of constraints $\boldsymbol{R}$, the iterative process will converge to $Q^*(x)$, a PD that satisfies *all* constraints in $\boldsymbol{R}$ and is closest to the original $P(X)$ measured by cross-entropy. Two difficulties exist here because IPFP works on the joint probability distributions, not on BNs. First, direct application of IPFP may destroy the existing interdependencies between variables (i.e., the given DAG becomes invalid). Secondly, IPFP is computationally very expensive because at *each* iteration, *every* entry in the joint PD of *all* variables in the BN must be updated. To overcome these difficulties, we developed an algorithm named **D-IPFP** [12, 2] which decomposes IPFP so that each iteration only updates one CPT of the given BN. In D-IPFP, Eq. (1) becomes: for each constraint $R(x_i \mid L_i)$ where L_i contains zero or more parents of x_i, the CPT of x_i is modified by.

$$Q_{(k)}(x_i \mid \pi_i) = Q_{(k-1)}(x_i \mid \pi_i) \cdot \frac{R(x_i \mid L_i)}{Q_{(k-1)}(x_i \mid L_i, LT)} \cdot \alpha_{k-1}(\pi_i), \tag{2}$$

where $\alpha_{k-1}(\pi_i)$ is the normalization factor. The process iterates over all $R(x_i \mid L_i)$ repeatedly until Q converges.

The translated BN preserves the semantics of the original ontology and is consistent with all the probabilistic constraints. It can support ontology reasoning tasks as probabilistic inferences in the translated BN. For example, given a concept description e, it can answer queries about concept satisfiability (whether $P(e|LT) = 0$) and about concept overlapping (as $P(|C_1, C_2| \, LT)$). It can also support semantic similarity measures such as Jaccard coefficient [14] and those based on information contents [13].

4. SLBN

When dealing with reasoning involving multiple BNs, existing approaches exchange beliefs via shared variables and impose very strong restrictions on how the shared variables are modeled in individual BNs [8]. *Semantically-Linked Bayesian Networks* (SLBN) are developed to support probabilistic inferences across independent developed BNs which do not share common variables but may have variables that have similar meaning or semantics [7].

Variable Linkage. Consider two concepts A in Onto1 and B in Onto2 with similar but not necessarily identical meaning. A and B become variables in BN1 and BN2, the two BNs translated from Onto1 and Onto2 by BayesOWL. We want to see the probabilistic inference being carried out from BN1 (the source) to BN2 (the destination). Note that BN1 and BN2 define two probability spaces, denoted PS^1 and PS^2. SLBN requires that the similarity information between A and B be given as the conditional distribution $P(B \mid A)$. This distribution is in yet another space, denoted as $PS^{1,2}$, which is related but different from PS^1 and PS^2. In particular, $PS^{1,2}$ shares variable A with PS^1 and B with PS^2. SLBN connects A to B with a directed variable linkage $L_B^A = < A, B, BN_A, BN_B, S_B^A >$, where $S_B^A = P(B \mid A)$ provides probabilistic information about the semantic similarity between A and B. The linkage L_B^A forms a pathway for A in BN_A to influence B in BN_B. However, since three separate probability spaces are involved, the Bayes' rule does not apply here. Instead, we use the Jeffrey's rule [3, 10]. This rule revises a PD $P(X)$ by another PD $Q(Y \subset X)$ over a subset of variables. The rule can be written as follows in the context of SLBN: to modify $P(X)$ by $Q(A)$ where $A \in X$, first, $P(A)$, the belief on A, changes to $Q(A)$,

$$P(A) \leftarrow Q(A). \tag{3}$$

Then the beliefs of other variables $B \in X$ are changed to

$$P(B) \leftarrow Q(B) = \sum_{a_i} P(B \mid A = a_i) Q(A = a_i) . \quad (4)$$

In BN literature, the probability such as $Q(A)$ is referred to as *soft* evidence about *A*, which is in contrast to the so-called *hard* evidence, e.g., $A = a_1$. Then, as depicted in Fig. 2, the influence to variables in PS^2 by *A* in PS^1 via the single linkage L_B^A can be viewed as twice applications of Jeffrey's rule across these three spaces, first from PS^1 to $PS^{1,2}$, then $PS^{1,2}$ to PS^2. In the first step, since variable *A* in PS^1 is identical to *A* in $PS^{1,2}$, $P(A)$ in PS^1 becomes soft evidence $Q(A)$ to $PS^{1,2}$, then the belief on *B* in $PS^{1,2}$ in the middle is updated to $Q(B) = \sum_{a_i} P(B \mid A) Q(A = a_i)$ by (4). In the second step, $Q(B)$ is then applied as soft evidence from $PS^{1,2}$ to variable *B* in PS^2, updating beliefs of other variables *C* in PS^2 as

$$Q(C) = \sum_{b_j} P(C \mid B = b_j) Q(b_j) = \sum_{b_j} P(C \mid B = b_j) \sum_{a_i} P(B = b_j \mid A = a_i) Q(A = a_i) .$$

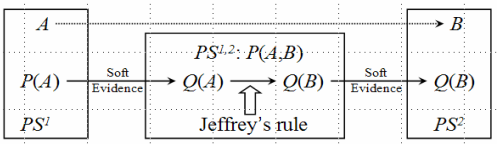


Figure 2: Variable *A* in BN_A influences *B* in BN_B via semantic linkage L_B^A.

Belief update with multiple linkages. When more than one variable linkage of semantically similar concepts exist, multiple soft evidences can be sent from one BN to the other via these linkages. One would naturally think of applying IPFP to this problem using all of the soft evidences as constraints. However, as discussed in Section 3, IPFP cannot be directly applied to BNs. To circumvent this difficulty we turn to another type of uncertain evidence, namely *virtual evidence* which is often given as a likelihood ratio $L(A = a_i) = (P(Ob(a_1) \mid a_i) : P(Ob(a_2) \mid a_i) : \cdots : P(Ob(a_n) \mid a_i))$ where $P(Ob(a_j) \mid a_i)$ is interpreted as the probability we observe *A* is in state a_j while *A* is actually in state a_i. One thing nice about virtual evidence is that it can be easily applied to BNs by adding a dummy or *virtual* node ve_A for the given $L(A)$. This node has no child, with *A* as its only parent, and its CPT is determined by $L(A)$ [9].

Soft evidence can be easily converted into virtual evidence when it is on a single variable [9]. A problem arises when multiple soft evidences, say $Q(A)$ and $Q(B)$, are converted to dummy nodes. Due to the interference, the results of belief update by the two virtual evidences will not confirm with either $Q(A)$ or $Q(B)$. What is needed is a method that can convert a set of soft evidences to likelihood ratios which, when all applied to the BN as virtual evidences, preserve every piece of soft evidence $Q(A)$. We have developed an algorithm for this by combining the virtual evidence and IPFP [8]. The page limit prevents a complete description of this algorithm, but it roughly works as follows. As an iterative process, it loops over the set of all soft evidences repeatedly until convergence. At each iteration *k*, only one soft evidence, say $Q(A)$, is picked up and a new virtual evidence node is added to the system with the likelihood ratio $L(A) = (Q(a_1) / P_{k-1}(a_i), \cdots, Q(a_s) / P_{k-1}(a_i))$ where $P_{k-1}(a_i)$ is from the BN with all virtual evidence nodes added in the previous $k-1$ iterations.

5. Learning Probabilities from the Web

The prior probabilities of non-root nodes in a BN can be computed from the CPT and the priors of root nodes, and, and there is usually only one root for BN translated from an ontology, the concept "THING", whose prior can either be assumed to 1 or estimated by some means or obtained from the ontology design. We will focus on learning conditional probabilities $P(C|D)$ for concepts *C* and *D*. Our approach is to use text classification techniques [1, 6] that builds classifiers for individual concepts by statistical analysis of the text exemplars associated with the concepts. Learning the probabilities for semantic similarity between concepts in two ontologies can be done through a cross-classification as follows. First, a statistical feature model (classifier) for each concept in Onto1 is built according to the statistical information in that concept's exemplars using a text classifier such as Rainbow [6]. Then concepts in Onto2 are classified into classes of Onto1 by feeding their respective exemplars into the models of Onto1 to obtain a set of scores, which can be interpreted as conditional probabilities for inter-concept similarity. Concepts in Onto1 can be classified in the same way into classes of Onto2. Conditional probabilities related to concepts in a single ontology can be obtained similarly through self-classification with the models learned for that ontology.

The performance of text classification based methods depends on the quality of exemplars attached to each concept. It is costly to find high quality text exemplars manually. Our approach is to use search engines such as Google to

retrieve text exemplars automatically from the web. The goal is to search for documents in which the concept is used in its intended semantics. The rationale is that the meaning of a concept can be described or understood by the way it is used. To search for documents relevant to a concept, one cannot simply use the words in the name of that concept as the key because a word may have multiple meanings. Fortunately, since we are dealing with concepts in well defined ontologies, the semantics of a concept is to a great extent specified by the other terms used in defining this concept in the ontology, including names of its superconcept classes and its properties. There are a number of ways to use the semantic information to improve search quality. A simple one that we have experimented is to form search query for a concept by combining all the terms on the path from root to that concept node in the taxonomy.

6. A Small Example

We have performed computer experiments on two small-scale real-world ontologies: the AI subdomain from ACM Topic Taxonomy (http://www.acm.org/class/1998/) and DMOZ (Open Directory http://dmoz.org/) hierarchy. These two hierarchies differ in both terminologies and modeling methods. DMOZ categorizes concepts to facilitate people's access to these pages, while ACM topic hierarchy categorizes concepts to structure a classification primarily for academics. For every concept, we obtained exemplars by querying Google and learned probability constraints as described in the Section 5. Then, *BayesOWL* is used to translate the two ontologies into two BNs as shown in Fig. 3.

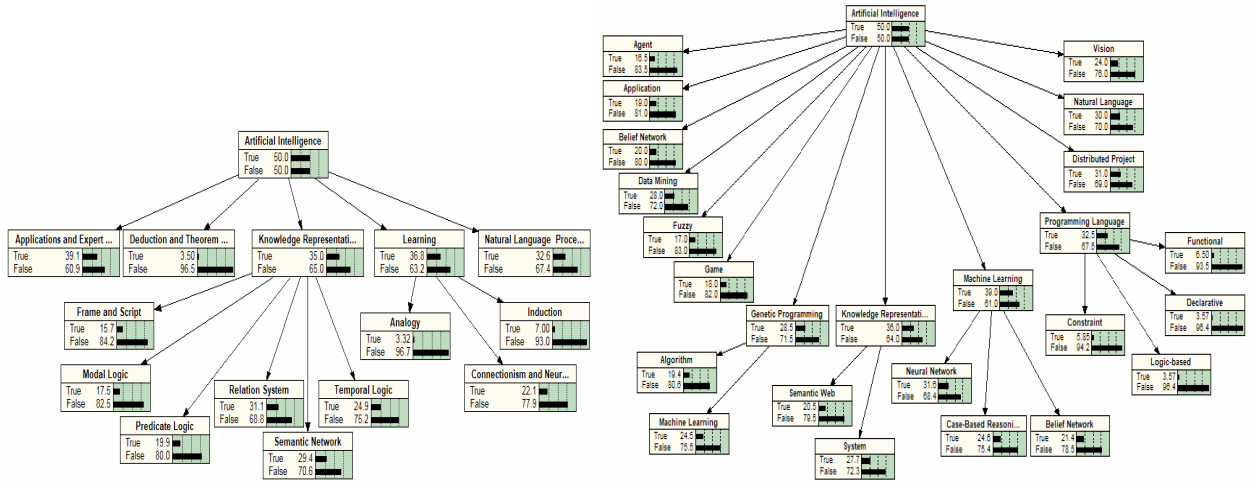


Figure 3: Two translated BN: from ACM (left) and DMOZ (right)

Joint distributions $P(A, B)$ were learned for each pair of concepts of the two BNs also by the Learner described in Section 5. Table 1 lists the five most similar concepts in the learning result, and their Jaccard coefficients computed from $P(A, B)$.

Table 1: Five most similar concepts in the learning result..

ACM Topic	**DMOZ**	**Similarity**
/Knowledge Rep. & Formalism Method	/Knowledge Representation	0.96
/Natural Language Processing	/Natural Language	0.90
/Learning	/Machine Learning	0.88
/Learning	/Knowledge Representation	0.81
/Applications & Expert System	/Knowledge Representation	0.79

Next, two variable linkages were created for the two pairs that are very similar. They are L_1 = < *dmoz.kr*, *acm.krfm*, BN_{dmoz}, BN_{acm}, S_1> and L_2 = < *dmoz.nl*, *acm.nlp*, N_{dmoz}, N_{acm}, S_2>, where

$$S_1 = P(acm.krfm \mid dmoz.kr) = \begin{pmatrix} 0.9943 & 0.0057 \\ 0.0973 & 0.9027 \end{pmatrix} \text{ and } \quad S_2 = P(acm.nlp \mid dmoz.nl) = \begin{pmatrix} 0.9843 & 0.0157 \\ 0.2327 & 0.7680 \end{pmatrix}$$

were calculated from their learned joint distributions.

SLBN allows us to conduct probabilistic reasoning well beyond finding the best concept matches. To illustrate our point, consider the example of finding a description of DMOZ's */Knowledge Representation/Semantic Web*

(*dmoz.sw*) in ACM topics. Apparently, there is no single ACM concept identical to *dmoz.sw*, the two most semantically similar concepts to *dmoz.sw* in ACM are

- /*Knowledge Representation and Formalism Method*/*Relation System* (*acm.rs*) and
- /*Knowledge Representation and Formalism Method*/*Semantic Network* (*acm.sn*)

with the learned joint distributions with Jaccard coefficients $J(dmoz.sw, acm.rs) = 0.64$, and $J(dmoz.sw, acm.sn) = 0.61$. The coefficient between *dmoz.sw* and *acm.krfm*, the super class of *acm.rs* and *acm.sn*, is even less (0.49). Most ontology mapping systems would stop here. However, with our framework, we can evaluate similarities with composite hypotheses involving multiple ACM concepts. One of such hypotheses is ***acm.rs*** Ú ***acm.sn***, which has Jaccard coefficient of **0.725**, significantly greater than any single concept candidate.

7. Conclusions

Our research has addressed a number of key issues of the probabilistic approach for ontology mapping. However, a few issues remain open, and a number of difficulties also need to be dealt with. Our BayesOWL is only completed for terminological taxonomies, it is not yet able to deal with properties. Similarly, our SLBN formalizes the notion of variable linkages to connect BNs and develops theoretically justifiable inference methods with such linkages. However, it does not address the important issue of how to determine whether a linkage should be established between a given pair of variables. Our learner for probabilities based on text classification and ontology guided search of the web is more problematic at this time. The probabilities generated by the learner may be inaccurate, and sometimes may also be inconsistent with each other. All these issues are potentially good topics for future research.

Acknowledgements

This work was supported in part by NSF award IIS-0326460 and NIST award 60NANB6D6206.

Disclaimer

Certain commercial software products are identified in this paper. This use does not imply approval or endorsement by NIST, nor does it imply that these products are necessarily the best available for the purpose.

References

1. Craven, M., et al, 1998, "Learning to extract symbolic knowledge from the World Wide Web", in Proc. of the 15th National Conference on Artificial Intelligence (AAAI-98), Madison, WI, 509 – 516.
2. Ding, Z., Peng, Y., and Pan, R., 2005, "BayesOWL: Uncertainty modeling in semantic web ontologies", in Soft Computing in Ontologies and Semantic Web, Z. Ma (Ed.) Springer-Verlag.
3. Jeffery, R. 1983. *The logic of Decisions*, 2nd Edition, University of Chicago Press.
4. Koller, D., Levy, A., and Pfeffer, A., 1997, "P-CLASSIC: A tractable probabilistic description logic", in Proc. of AAAI-97, 390-397.
5. Kruithof, R., 1937, "Telefoonverkeersrekening", De Ingenieur 52:E15-E25.
6. McCallum, A., 1996, "Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering", http://www.cs.cmu.edu/~mccallum/bow.
7. Pan, R., Ding, Z., Yu, Y. and Peng, Y., 2005, "A Bayesian Network Approach to Ontology Mapping", in Proc. of the Fourth International Semantic Web Conference, Nov. 6-10, Galway, Ireland.
8. Pan, R., Peng, Y., and Ding, Z., 2006, "Belief Update in Bayesian Networks Using Uncertain Evidence", in Proc. of the IEEE International Conf. on Tools with Artificial Intelligence, Nov. 13 – 15, Washington, DC.
9. Pearl, J., 1988, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kauffman Publishers.
10. Pearl, J. 1990, "Jeffrey's rule, passage of experience, and neo-Bayesianism", in H.E. Kyburg, et al. (eds.), Knowledge Representation and Defeasible Reasoning, 245-265.
11. Peng, Y., et al, 2003, "Semantic Resolution for E-Commerce", in Innovative Concepts for Agent-Based Systems, Springer-Verlag, 355-366.
12. Peng, Y. and Ding, Z., 2005, "Modifying Bayesian Networks by Probability Constraints", in Proc. of 21st Conference on Uncertainty in Artificial Intelligence, July 26-29, Edinburgh, Scotland.
13. Resnik, P., 1995, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy", in Proc. of the 14th Intl. Joint Conf. on AI, 448-453, Montreal, CA
14. van Rijsbergen, C. J., 1979, Information Retrieval. London: Butterworths. Second Edition.