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 modules. 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

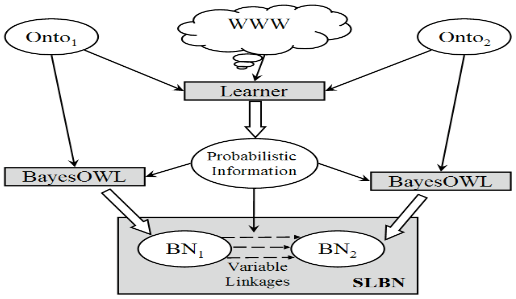
1. Uncertainty in Ontology Mapping and Semantic Integration

Representing and reasoning with uncertainty has 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 degree 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, the semantic web ontology language. Figure 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 given in the next three sections.



- *BayesOWL* translates tw 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 prior and conditional probabilities about the classes and superclass relations defined in the ontology called probabilistic constraints to 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. If classes A and B represent two concepts in an ontology, we treat them as random binary variables and interpret $P(A = a)$ as the prior probability 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. 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”, “CondProb” for their eoncoding. 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.)

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 ($i = 1,\dots,n$) 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 $C1$ and $C2$ 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 $C1$ and $C2$ to the L-Node.

Constructing CPT. The nodes in the DAG 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 a L-Node in X_L can be determined by the logical relation it represents so that when its state 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) to construct CPTs for concept nodes. IPFP [2], first published in 1937 [5], is a mathematical procedure that modifies a given probability distribution $P(X)$ 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, the process starts with $Q_0(x) = P(X)$, the initial distribution, and at each successive iteration it modifies the distribution 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 if all constraints in $\boldsymbol{R}$ are consistent, then the iterative process will converge to $Q^*(x)$, a distribution 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 since every entry in the joint distribution of the entire BN must be updated at each iteration. To overcome these difficulties, we developed an algorithm named **D-IPFP** [12,2] that 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 parent 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 common 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), about concept overlapping (how close *e* is to a concept *C* as *P(e*|*C, LT)*). It 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 constraints on the construction of individual BN. *Semantically-Linked Bayesian Networks* (SLBN) is 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* and *B* defined in Onto1 and Onto2, respectively, 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 $P(A \mid B)$. This distribution is in yet another probability 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. We use a directed pair-wise variable linkage to link two semantic similar variables from the source BN to the destination BN. Specifically, a pair-wise variable linkage L_B^A from variable *A* in network BN_A to variable *B* in network BN_B is defined as $< A, B, BN_A, BN_B, S_B^A >$, where S_B^A, the conditional probability of *B*, given *A*, quantifies the semantic similarity between *A* and *B*.

The linkage L_B^A provides 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 distribution $P(X)$) by another distribution $Q(Y \subset X)$ 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, is modified to $Q(A)$,

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

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

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

In BN literature, the probability information such as $Q(A)$ in evidential reasoning 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 semantically 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 by (4) to $Q(B) = \sum_{a_i} P(B \mid A)Q(A = a_i)$. 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 by (4) as

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

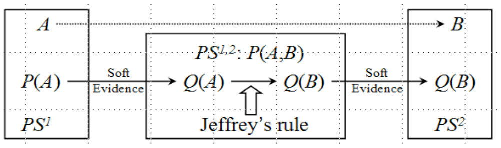


Figure 2. Variable A in BN_A influences B in BN_B via semantic linkage L_B^A

Belief propagation with multiple linkages. When more than one variable linkages of semantically similar concepts exist, multiple soft evidences can be sent from the source BN to the destination BN via these linkages. One would naturally think of applying IPFP to this problem using all of the soft evidence as constraints. However, as discussed in Section 3, IPFP cannot be directly applied to BNs. For small BN, one can explicitly generate the full joint distribution from the BN and then apply IPFP for belief update on the distribution. This is infeasible for large BN because the distribution would be too big. To address these difficulties we turn to another type of uncertain evidence, namely *virtual evidence* which is often given as a likelihood ratio, in the form of $L(A) = (P(Ob(a_1) \mid a_1) : P(Ob(a_2) \mid a_2) : \cdots : P(Ob(a_n) \mid a_n))$ where $P(Ob(a_i) \mid a_i)$ is interpreted as the probability we observe A is in state a_i if A is indeed in state a_i . One thing nice of virtual evidence is that it can be easily applied to BN 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 ration $L(A) = (Q(a_1)/P_{k-1}(a_1),...,Q(a_s)/P_{k-1}(a_s))$ where $P_{k-1}(a_i)$ is the distribution with all virtual evidence nodes added in the previous $k-1$ iterations.

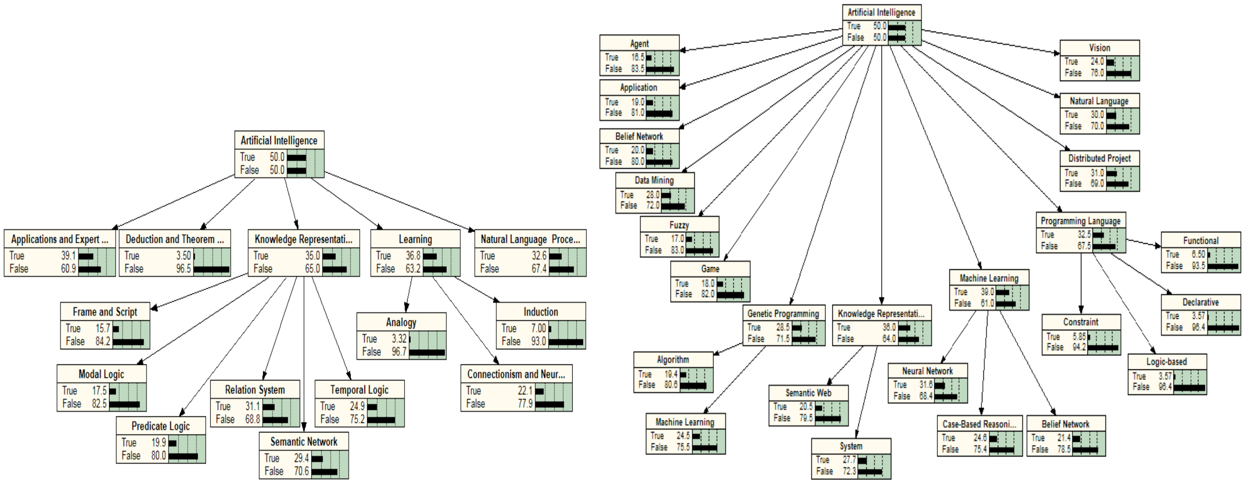
5. Learning Probabilities From The Web

In this framework, we use priors $P(C)$ to specify the uncertainty about concept C, conditionals $P(C|D)$ for relations between concepts C and D. Often these kinds of probabilistic information are not available and are difficult to obtain from domain experts. Our solution is to learn them using text classification technique [1, 6] which 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. Similarly, prior and 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 the semantic information can be used 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 and DMOZ[1] (Open Directory) 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 Figure 3.



Artificial Intelligence True 50.0 False 50.0
Applications and Expert ... True 39.1 False 60.9
Deduction and Theorem ... True 3.50 False 96.5
Knowledge Representati... True 35.0 False 65.0
Learning True 36.8 False 63.2
Natural Language Proce... True 32.6 False 67.4
Frame and Script True 15.7 False 84.2
Analogy True 3.32 False 96.7
Induction True 7.00 False 93.0
Modal Logic True 17.5 False 82.5
Relation System True 31.1 False 68.8
Temporal Logic True 24.9 False 75.2
Connectionism and Neur... True 22.1 False 77.9
Predicate Logic True 19.9 False 80.0
Semantic Network True 29.4 False 70.6
Artificial Intelligence True 50.0 False 50.0
Agent True 16.5 False 83.5
Application True 19.0 False 81.0
Belief Network True 20.0 False 80.0
Data Mining True 28.0 False 72.0
Fuzzy True 17.0 False 83.0
Game True 18.0 False 82.0
Vision True 24.0 False 76.0
Natural Language True 30.0 False 70.0
Distributed Project True 31.0 False 69.0
Programming Language True 32.5 False 67.5
Functional True 6.50 False 93.5
Declarative True 3.57 False 96.4
Constraint True 5.85 False 94.2
Logic-based True 3.57 False 96.4
Machine Learning True 39.0 False 61.0
Genetic Programming True 28.5 False 71.5
Knowledge Representati... True 36.0 False 64.0
Algorithm True 19.4 False 80.6
Machine Learning True 24.5 False 75.5
Semantic Web True 20.5 False 79.5
System True 27.7 False 72.3
Neural Network True 31.6 False 68.4
Case-Based Reasoni... True 24.6 False 75.4
Belief Network True 21.4 False 78.5

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

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

were calculated from their learned joint distributions.

[1] http://www.acm.org/class/1998/ and http://dmoz.org/

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* ∨ *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. Due to the uneven quality of the search, the probabilities generated by the learner not only may be inaccurate, but sometime 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.

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 Proceedings of the IEEE International Conference 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 21^{st} 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.