BayesOWL: A Probabilistic Framework for Uncertainty in Semantic Web

Zhongli Ding
Ph.D. Dissertation Defense

Academic Adviser:
Dr. Yun Peng

December 5, 2005

A Brief Bio of Zhongli Ding

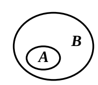
- The Long Path
  - Grown up in south China
  - Earned B.S. in July 1999 from USTC, Hefei
  - Started graduate study in September 1999
  - Earned M.S. in May 2001 from UMBC
  - Decided to continue the study in September 2001
  - Advanced to candidacy in July 2003
- Publications (partial list)
  - BayesOWL: Uncertainty Modeling in Semantic Web Ontologies (Book Chapter)
  - A Bayesian Network Approach to Ontology Mapping (ISWC 2005)
  - Modifying Bayesian Networks by Probability Constraints (UAI 2005)
  - A Bayesian Methodology towards Automatic Ontology Mapping (AAAI C&O'05)
  - A Bayesian Approach to Uncertainty Modeling in OWL Ontology (AISTA-2004)
  - A Probabilistic Extension to Ontology Language OWL (HICSS-37)

Outline

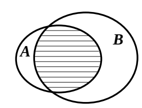
- Motivation
- Thesis Statement
- Brief Background
- Modifying CPTs of BNs by Probabilistic Constraints
- BayesOWL – A Probabilistic Extension to OWL
- Conclusion and Future Research

Motivation

- Partial or incomplete knowledge
  - degree of inclusion
  - degree of overlap
  - degree of closeness
  - degree of similarity
  - noisy or uncertain input in reasoning
  - uncertainty in concept mapping (loss of information)
- Probability theory and Bayesian networks



$\Pr(a \mid \bar{b}) = 0$ and $\Pr(a \mid b) = \frac{|A|}{|B|} = 0.1$



$\Pr(b \mid a) = \frac{|A \cap B|}{|A|} = 0.9$

Thesis Statement

None of the existing ontology languages, including OWL, provides a means to quantify the degree of overlap or inclusion between concepts, or to support reasoning with uncertain information and knowledge in ontology engineering.

This work is an attempt to develop a framework which augments and supplements OWL with additional expressive power for representing and reasoning with uncertainty based on Bayesian networks (BNs).

Brief Background – 1
Semantic Web, Ontology, Bayesian Networks

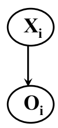
- Semantic Web: Provides semantics for information exchanged over Internet.
- Ontology: An explicit specification of a conceptualization.
  - RDF(S) and OWL: Define concepts and relations about concepts in a particular domain.
  - Description Logics (DLs): Provide decidable and sound inference mechanism.
- Bayesian Networks (BNs): DAG + CPT
  - Chain Rule:

$$P(x_1,\ldots,x_n) = \prod_i P(x_i \mid \pi_i)$$

7/65

Brief Background – 2
Evidences in BNs

- Hard Evidence: Instantiates a node to a particular state.
  - i.e., $Pr(X_i = x_i) = 1$ and $Pr(X_i = x_i' \neq x_i) = 0$
- Soft Evidence: Gives a distribution of a node on its states.
  - hard evidence is a special case of soft evidence
- Virtual Evidence: The likelihood of a variable's distributions.
  - the probability of observing X_i being in state x_i if its true state is x_i'
  - virtual evidence is equivalent to soft evidence in expressiveness



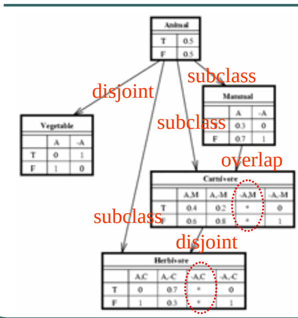
8/65

Brief Background – 3
Works on Extending DLs with BNs – 1

- P-CLASSIC (can not handle "Equivalence" operator)
- PTDL (TDL includes only "Conjunction" and "Role Quantification" operators)
- PR-OWL extends OWL with full first-order expressiveness based on MEBNs
- OWL_QM extends OWL to support the representation of PRMs
- Holi and Hyvönen uses BNs to model degrees of conceptual overlap only for ontologies encoded in RDF(S)

9/65

Brief Background – 4
Works on Extending DLs with BNs – 2



DAG construction in P-CLASSIC is arbitrary, no specific rules to follow.

Sometimes it is difficult or even impossible for P-CLASSIC to assign CPTs to some nodes.

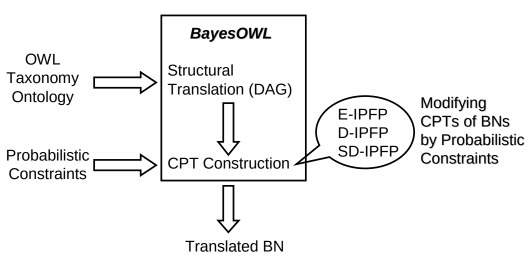
10/65

Brief Background – 5
Works on Representing Probabilistic Information

- Fukushige proposes an extended vocabulary for describing probabilistic relations in a way that is both semantic web compatible and easy to map to a BN.
- The OWL_QM system includes an OWL implementation of PRM constructs about facets, slot chains, an owl class named "ProbabilisticRelationship" for casual relations, and a set of vocabularies to define the probability distributions or tables in a probabilistic relationship.

11/65

My Work



12/65

Modifying CPTs of BNs by Probabilistic Constraints

- Preliminary
- The Problem
- E-IPFP
- D-IPFP
- SD-IPFP
- The *IPFP* API
- Experiments
- Summary

Preliminary – 1
I-divergence, Total Variance, Dominance, I_1-projection

I-divergence

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

Total Variance

$$|P - Q| = \sum_{x \in X} |P(x) - Q(x)|$$

Dominance

$P \ll Q$ means $\{x \in X | P(x) > 0\} \subseteq \{y \in X | Q(y) > 0\}$
$\{y \in X | Q(y) = 0\} \supseteq \{x \in X | P(x) = 0\}$

I_1-projection: Let $\mathfrak{P}$ be the set of JPDs over $X = \{X_1, ..., X_n\}$. The I_1-*projection* of a JPD $Q \in \mathfrak{P}$ on a given set of JPDs $\boldsymbol{\varepsilon} \subseteq \mathfrak{P}$ is a JPD $P \in \boldsymbol{\varepsilon}$ such that the *I-divergence* "$I(P\|Q)$" is minimal among all JPDs in $\boldsymbol{\varepsilon}$.

Preliminary – 2
IPFP (Iterative Proportional Fitting Procedure), C-IPFP

IPFP

$$Q_{(k)}(X) = \begin{cases} 0 & if\ Q_{(k-1)}(Y^i) = 0 \\ Q_{(k-1)}(X) \cdot \frac{R_i(Y^i)}{Q_{(k-1)}(Y^i)} & if\ Q_{(k-1)}(Y^i) > 0 \end{cases}$$

Modify the joint distribution to meet ***one*** constraint at each step.

C-IPFP

$$Q_{(k)}(X) = \begin{cases} 0 & if\ Q_{(k-1)}(Y^i|Z^i) = 0 \\ Q_{(k-1)}(X) \cdot \frac{R_i(Y^i|Z^i)}{Q_{(k-1)}(Y^i|Z^i)} & if\ Q_{(k-1)}(Y^i|Z^i) > 0 \end{cases}$$

Converge to distribution Q*, which is an I_1-projection of the initial distribution $Q_{(0)}$.

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

The Problem – 1

- Modifying CPTs of a BN to meet a set of given low-dimensional probabilistic constraints is important when
  - designing new BNs
  - merging small BNs into a large one
  - refining an existing BN with new or more reliable probabilistic information

  because it is not that easy to obtain the CPTs for each of the variables since experts' opinions are often
  - coarse
  - not in a uniform scale
  - not in the form of CPTs

- Learning CPTs from statistical data usually requires samples of complete instantiations of all variables

The Problem – 2

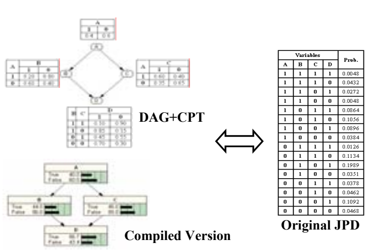
$\mathcal{R}_m$ denotes a set of marginal constraints $\{R_i(Y^i)\}$. $\mathcal{R}_c$ denotes a set of conditional constraints $\{R_i(Y^i|Z^i)\}$. $\mathcal{R} = \mathcal{R}_m \cup \mathcal{R}_c$ denotes a set of constraints, either marginal, or conditional.

Problem Definition:

for a given $\mathbf{N}$ over variables X with distribution $Q(X)$ and a set of consistent constraints $\mathcal{R}$, find $\mathbf{N}^*$ that meets the following three requirements:

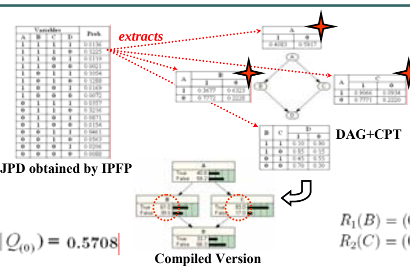
1. $\mathcal{G} = \mathcal{G}^*$ (both networks have the same structure);
2. $Q^*(X)$, the distribution of $\mathbf{N}^*$, satisfies all constraints in $\mathcal{R}$; and
3. *I-divergence* $I(Q^*(X)\|Q(X))$ is minimum among all distributions that meet requirements 1 and 2.

The Problem – 3



Network $\mathbf{N}_1$ of $X = \{A, B, C, D\}$ and its CPTs

The Problem – 4

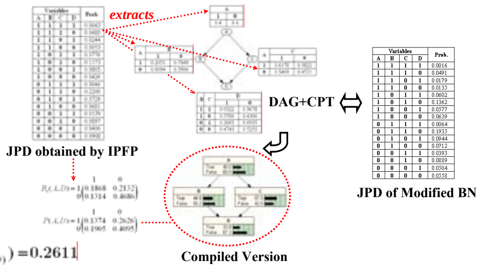


$I(Q^* \| Q_{(0)}) = 0.5708$

$R_1(B) = (0.61, 0.39)$
$R_2(C) = (0.83, 0.17)$

Running *IPFP* with $R_1(B)$ and $R_2(C)$

The Problem – 5



$I(Q^* \| Q_{(0)}) = 0.2611$

$Q^*(A,B,C,D) \neq Q^*(A) \cdot Q^*(B|A) \cdot Q^*(C|A) \cdot Q^*(D|B,C)$

Running *IPFP* with $R_3(A,D)$

E-IPFP – 1

$Q_0(X)$

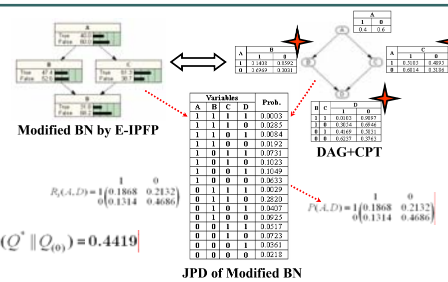
- Extend IPFP to handle requirement 1: the solution distribution should be consistent with the network structure, i.e., $Q^*(X) = \prod_{i=1}^{n} Q^*(X_i \mid \pi_i)$
- Treat this requirement as the (m+1)th probabilistic constraint to IPFP, name it "structural constraint", i.e.,

  $R_{m+1}(X)$: $R_r(X) = \prod_{i=1}^{n} Q_{(k-1)}(X_i|\pi_i)$
- We can prove that this process converges to a distribution $Q^*(X)$ and $Q^*(X)$ is an I_1-projection on constraint set $\{R_1, R_2, \ldots, R_m, R_{m+1}(X)\}$

E-IPFP – 2

```
E-IPFP(N_0(X), R = {R_1, R_2, ..., R_m}) {
1. Computing Q_0(X) = ∏_{i=1}^n Q_0(X_i|π_i) where Q_0(X_i|π_i) ∈ C_0;
2. Starting with k = 1, repeat the following procedure until convergence {
    2.1. i = ((k − 1) mod (m + 1)) + 1;
    2.2. if i < m + 1 {
            if (R_i ∈ R_m) { /* for marginal constraints */
               Q_(k)(X) = Q_(k−1)(X) · R_i(Y^i) / Q_(k−1)(Y^i)      ; (from Eq. 2.6)
            } else if (R_i ∈ R_c) { /* for conditional constraints */
               Q_(k)(X) = Q_(k−1)(X) · R_i(Y^i|Z^i) / Q_(k−1)(Y^i|Z^i)   ; (from Eq. 2.7)
            }
         }
    2.3. else {
            extract Q_(k−1)(X_i|π_i) from Q_(k−1)(X) according to G_0;
            Q_(k)(X) = ∏_{i=1}^n Q_(k−1)(X_i|π_i);
         }
    2.4. k = k + 1;
   }
3. return N*(X) with G* = G_0 and C* = {Q_(k)(X_i|π_i)};
}
```


E-IPFP – 3



$I(Q^* \| Q_{(0)}) = 0.4419$

Running *E-IPFP* with $R_3(A,D)$

D-IPFP – 1

- The computation of both IPFP and E-IPFP is on the entire joint distribution of ***X*** at every iteration → expensive, i.e.,

  $O(2^n * 2^{|Y^i|})$ for priors or $O(2^n * 2^{|Y^i|+|Z^i|})$ for conditionals.
- Can we utilize the interdependencies imposed on the distribution by the network structure and only update some selected CPTs?
- Decomposes the *global* E-IPFP (the one involving all ***n*** variables) into a set of *local* E-IPFP, each for one constraint $R_i(Y^i)$ (or $R_i(Y^i|Z^i)$), on a small subnet of $\mathcal{N}$ that contains Y^i (or $Y^i \cup Z^i$), i.e., $R_i(Y^i)$ (or $R_i(Y^i|Z^i)$) is used to modify $Q_{(k-1)}(Y|S)$.

D-IPFP – 2

```
D-IPFP(N_0(X), R = {R_1, R_2, ..., R_m}) {
1. Computing Q_0(X) = ∏_{i=1}^n Q_0(X_i|π_i) where Q_0(X_i|π_i) ∈ C_0;
2. Starting with k = 1, repeat the following procedure until convergence {
    2.1 i = ((k − 1) mod m) + 1;
    2.2 getting the Y and S of this constraint, let counter kk = 0;
    2.3 while not converge {
        2.3.1 Q'_(k−1)(Y|S) = ∏_{X_j∈Y} Q_(k−1)(X_j|π_j) ;
        2.3.2 if (R_i ∈ R_m) { /* for marginal constraints */
                 Q'_(k)(Y|S) = Q'_(k−1)(Y|S) · R_i(Y^i) / Q'_(k−1)(Y^i) · α_k;
              } else if (R_i ∈ R_c) { /* for conditional constraints */
                 Q'_(k)(Y|S) = Q'_(k−1)(Y|S) · R_i(Y^i|Z^i) / Q'_(k−1)(Y^i|Z^i) · α_k;
              }
        2.3.3 Updating CPTs for all X_j ∈ Y and compiling the network; {
                 Q_(k)(X_j|π_j) = Q'_(k)(X_j|π_j)   ∀X_j ∈ Y
                 Q_(k)(X_j|π_j) = Q_(k−1)(X_j|π_j)  ∀X_j ∉ Y
              }
        2.3.4 Q_(k−1)(X_j|π_j) = Q_(k)(X_j|π_j), for all X_j ∈ X ; kk = kk + 1 ;
       }
    2.4 k = k + 1;
   }
3. return N*(X) with G* = G_0 and C* = {Q_(k)(X_i|π_i)};
}
```

$Y \subseteq X$ **and** $Y \neq \emptyset$ **and** $Y^i \subseteq Y$

$S = (\cup_{X_j \in Y} \pi_j) \backslash Y$

How to get Y?

(Y1) Y=X, S=∅

(Y2) Y=Y^i (or Y=Y^i ∪ Z^i for conditionals)

(Y3) Get initial Y and S as (Y2), then repeat the following process until nothing more could be added to Y:

if $X_i \in S$ and $X_j \in Y^i$ and $X_j \in \pi_i$, then

$Y = Y \cup \{X_i\}$,

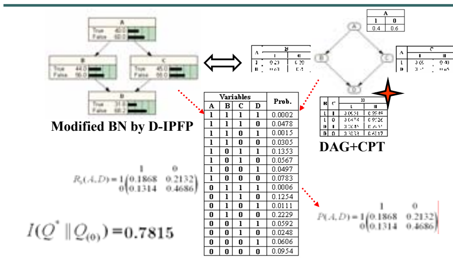
$S = (S - \{X_i\}) \cup (\pi_i \backslash Y)$

D-IPFP – 3

- We can prove that:
  - each iteration of D-IPFP for one constraint produces an I_1-projection of the previous distribution on a constraint set $\{R_i(Y^i), R_r(Y)\}$
  - the complete D-IPFP process converges to a distribution $Q^*(X)$ which is an I_1-projection of $Q_0(X)$ on constraint set $\mathcal{R} \cup \{\forall R_r(Y)\} \cup \{R_r(X)\}$
- Sacrifice I-divergence for significant saving in computation: $O(2^{n-|S|-|Y|})$

$$O(2^n * 2^{|Y^i|}) \rightarrow O(2^{|S|+|Y|} * 2^{|Y^i|})$$

D-IPFP – 4



Variables				Prob.
A	B	C	D	
1	1	1	1	0.0002
1	1	1	0	0.0478
1	1	0	1	0.0015
1	1	0	0	0.0305
1	0	1	1	0.1353
1	0	1	0	0.0567
1	0	0	1	0.0497
1	0	0	0	0.0783
0	1	1	1	0.0006
0	1	1	0	0.1254
0	1	0	1	0.0111
0	1	0	0	0.2229
0	0	1	1	0.0592
0	0	1	0	0.0248
0	0	0	1	0.0606
0	0	0	0	0.0954

JPD of Modified BN

Running *D-IPFP*(Y2) with $R_3(A, D)$

SD-IPFP – 1

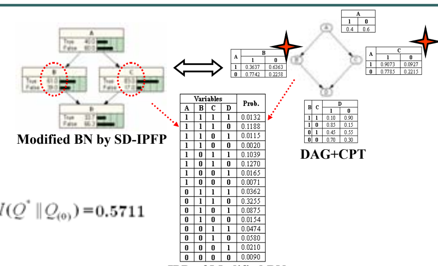
A probabilistic constraint is *local* if it contains only one variable and some of its parents. Thus,

$$Y = \{X_j\},\ S = \pi_j$$

$$\begin{cases} Q_{(k)}(X_j|\pi_j) = Q_{(k-1)}(X_j|\pi_j) \cdot \frac{R_i(Y^i)}{Q_{(k-1)}(Y^i)} \cdot \alpha_k \\ Q_{(k)}(X_l|\pi_l) = Q_{(k-1)}(X_l|\pi_l) \quad for\ l \neq j \end{cases}$$

Similar equations can be obtained in case of conditional constraint.

SD-IPFP – 2



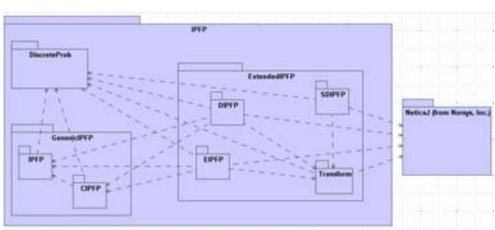
A	B: 1	B: 0
1	0.3637	0.6363
0	0.7742	0.2258

Variables				Prob.
A	B	C	D	
1	1	1	1	0.0132
1	1	1	0	0.1188
1	1	0	1	0.0115
1	1	0	0	0.0020
1	0	1	1	0.1039
1	0	1	0	0.1270
1	0	0	1	0.0165
1	0	0	0	0.0071
0	1	1	1	0.0362
0	1	1	0	0.3255
0	1	0	1	0.0875
0	1	0	0	0.0154
0	0	1	1	0.0474
0	0	1	0	0.0580
0	0	0	1	0.0210
0	0	0	0	0.0090

JPD of Modified BN

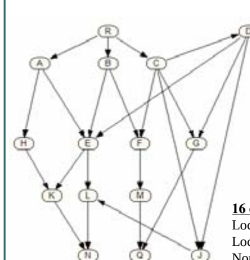
Running *SD-IPFP* with $R_1(B)$ and $R_2(C)$

The *IPFP* API



Class Diagram of the *IPFP* API

Experiments – 1



4 constraints: (2 local ones + 2 non-local ones)
Local Marginal (1): {P(B)}
Local Conditional (1): {P(G|C)}
Non-local Marginal (1): {P(R,D)}
Non-local Conditional (1): {P(J|R)}

8 constraints: (5 local ones + 3 non-local ones)
Local Marginal (3): {P(B), P(C), P(H)}
Local Conditional (2): {P(F|C), P(N|K)}
Non-local Marginal (1): {P(R,D)}
Non-local Conditional (2): {P(N|R,D), P(Q|R,D)}

16 constraints: (10 local ones + 6 non-local ones)
Local Marginal (5): {P(A), P(B), P(C), P(H), P(M)}
Local Conditional (5): {P(E|B,D), P(F|B), P(F|C), P(G|C), P(N|K)}
Non-local Marginal (3): {P(R,D), P(K,L), P(N,Q)}
Non-local Conditional (3): {P(J|R), P(N|R,D), P(Q|R,D)}

Experiments – 2

Num. of Constraints	Num. of Iterations			Time Used (seconds)			Total Variance $\|Q^* - Q_0\|$			I-divergence $I(Q^* \|\| Q_0)$		
	E-IPFP	D-IPFP (Y2)	D-IPFP (Y3)	E-IPFP	D-IPFP (Y2)	D-IPFP (Y3)	E-IPFP	D-IPFP (Y2)	D-IPFP (Y3)	E-IPFP	D-IPFP (Y2)	D-IPFP (Y3)
4	8	3	7	533	0.75	0.406	0.2757	0.4113	0.2785	0.0811	0.3401	0.0826
8	8	9	4	581	107	15	0.6192	0.6694	0.6226	0.5618	0.6292	0.5665
16	9	7	5	769	45	189	1.2579	1.2595	1.2593	2.5284	2.6146	2.5321

Summary

- What have already been done
  - E-IPFP extends IPFP to modify probability distributions represented as BNs
  - Significant savings in computational cost by D-IPFP which decomposes the global E-IPFP into local ones with a much smaller scale
  - D-IPFP is simplified and rewritten to SD-IPFP when only dealing with local constraints (priors or pair-wise marginals)
  - SD-IPFP is further extended in ***BayesOWL*** for CPT construction under the condition of a given set of hard evidences
- What else could be done
  - How to improve efficiency
    - The order of applying constraints may further reduce the speed
    - Divide a large constraint into smaller ones by exploring independence between the variables (possibly based on the network structure)
    - Other optimizations, such as parallelizing, approximation, etc.
  - How to handle inconsistent constraints
  - In what situations the modification of only CPTs is no longer sufficient or desirable and the DAG need also be changed in order to better satisfy given constraints

BayesOWL – A Probabilistic Extension to OWL

- Representing Probabilistic Information
  - Not supported by current OWL
  - Define new classes for prior and conditional probabilities
- Structural Translation
  - Class hierarchy: set theoretic approach
  - Logical relations (equivalence, complement, disjoint, union, intersection): introducing L-Nodes
- CPT Construction
  - SD-IPFP
- Semantics of ***BayesOWL***
  - Preserve semantics of the original ontology
  - Encoded probability distributions among relevant variables
- Reasoning
  - Concept Satisfiability
  - Concept Overlapping
  - Concept Subsumption
- The *OWL2BN* API
- Comparison to Existing Works
- Summary

Representing Probabilistic Information – 1

- Concept ***C*** is mapped to a binary random variable with two states, ***c*** and $\bar{c}$
- ***P(C = c)*** is interpreted as the prior probability or one's belief that an arbitrary individual belongs to class ***C***
- ***P(C = c | D = d)*** is interpreted as the conditional probability that an individual of class ***D*** also belongs to class ***C***
- Two kinds of probabilistic information / constraints
  - Prior or marginal probability ***P(C)***: for classes
  - Conditional probability $P(C|O_C)$, where $O_C \subseteq \pi_C$, $\pi_C \neq \emptyset$, $O_C \neq \emptyset$: especially for pair-wise conditionals for RDF triples
- Three new OWL classes: "PriorProb", "CondProb", "Variable"
  - PriorProb: "hasVarible" (1), "hasProbValue" (1)
  - CondProb: "hasCondition" (1 or more), "hasVariable" (1), "hasProbValue" (1)
  - Variable: "hasClass" (1), "hasState" (1)

Representing Probabilistic Information – 2

- **Example 1: *P*(*c*) = 0.8**

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

- **Example 2: *P*(*c*|*p1*,*p2*,*p3*) = 0.8**

```
<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>
```

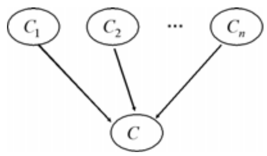

Structural Translation – 1

Constructor	DL Syntax	Class Axiom	Logical Operator
rdfs:subClassOf	$C_1 \sqsubseteq C_2$	*	
owl:equivalentClass	$C_1 \equiv C_2$	*	
owl:disjointWith	$C_1 \sqsubseteq \neg C_2$ and $C_2 \sqsubseteq \neg C_1$	*	
owl:unionOf	$C \equiv C_1 \sqcup \ldots \sqcup C_n$		*
owl:intersectionOf	$C \equiv C_1 \sqcap \ldots \sqcap C_n$		*
owl:complementOf	$\neg C$		*

Supported Constructors

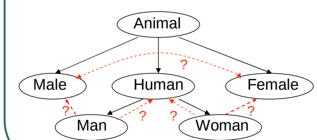
Structural Translation – 2

- General Principle
  - Set-theoretic approach: each concept class treated as a set of objects/instances
  - Every primitive or defined concept class ***C***, is mapped into a two-state (either "**True**" or "**False**") variable node in the translated BN, called it **concept node**
  - An directed arc is drawn from a superclass node to a subclass node



Structural Translation – 3

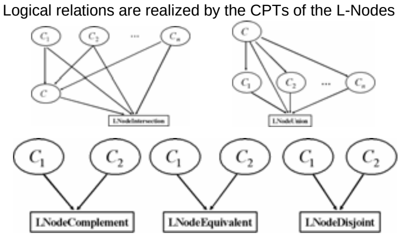
- Pure concept subsumption hierarchy can be easily translated into BN based on subclass relations in OWL.
- **An example**
  - "Animal" is a primitive concept class
  - "Male", "Female", and "Human" are three subclasses of "Animal"
  - "Man" and "Woman" are two subclasses of "Human"



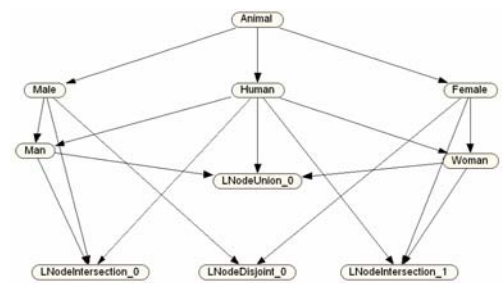
- **What about other logical relations?**
  - "Male" and Female" are disjoint with each other
  - "Man" is an intersection of "Human" and "Male"
  - "Woman" is an intersection of "Human" and "Female"
  - "Human" is the union of "Man" and "Woman"
- **Difficult to model these relations by probabilistic interdependencies among the related variables**
  - intersection
  - union
  - complement
  - equivalent
  - disjoint

Structural Translation – 4

- Create **L-Node**s, one for each defined logical relation
  - Each concept node in the relation has an arc pointing to the L-Node
  - Logical relations are realized by the CPTs of the L-Nodes



Structural Translation – 5

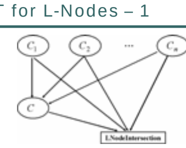


CPT Construction – 1

- Two kinds of nodes:
  - X_C: concept nodes for regular concept classes
    - ***P(C)*** or ***P(C|O_C)***, where $O_C \subseteq \pi_C$, $\pi_C \neq \emptyset$, $O_C \neq \emptyset$
    - may impose additional conditional independence to the concept nodes by the d-separation in the BN structure, which can be viewed as default relationships, unless information to the contrary is provided
  - X_L: L-Nodes for bridging concept nodes that are associated by logical relations
    - L-Nodes are leaf nodes, with only in-arcs, which help avoid forming cycles in the translated DAG
    - logical relations are separated from the "rdfs:subClassOf" relation, which makes CPTs smaller and easier to construct
    - when the states of all L-Nodes are "**True**" (named it τ), the logical relations in the ontology will be held

CPT Construction – 2

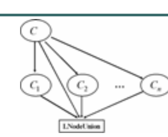
CPT for L-Nodes – 1



C1	C2	C	True	False
True	True	True	1.000	0.000
True	True	False	0.000	1.000
True	False	True	0.000	1.000
True	False	False	1.000	0.000
False	True	True	0.000	1.000
False	True	False	1.000	0.000
False	False	True	0.000	1.000
False	False	False	1.000	0.000

LNodeIntersection = True iff

$cc_1c_2 \vee \bar{c}\bar{c}_1c_2 \vee \bar{c}c_1\bar{c}_2 \vee \bar{c}\bar{c}_1\bar{c}_2$



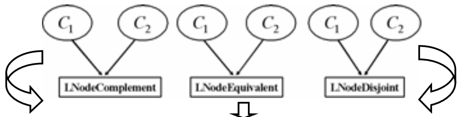
C1	C2	C	True	False
True	True	True	1.000	0.000
True	True	False	0.000	1.000
True	False	True	1.000	0.000
True	False	False	0.000	1.000
False	True	True	1.000	0.000
False	True	False	0.000	1.000
False	False	True	0.000	1.000
False	False	False	1.000	0.000

LNodeUnion = True iff

$cc_1c_2 \vee c\bar{c}_1c_2 \vee cc_1\bar{c}_2 \vee \bar{c}\bar{c}_1\bar{c}_2$

CPT Construction – 3

CPT for L-Nodes – 2



C1	C2	True	False
True	True	0.000	1.000
True	False	1.000	0.000
False	True	1.000	0.000
False	False	0.000	1.000

C1	C2	True	False
True	True	1.000	0.000
True	False	0.000	1.000
False	True	0.000	1.000
False	False	1.000	0.000

C1	C2	True	False
True	True	0.000	1.000
True	False	1.000	0.000
False	True	1.000	0.000
False	False	1.000	0.000

LNodeComplement = True iff $c_1\bar{c}_2 \vee \bar{c}_1 c_2$

LNodeEquivalent = True iff $c_1 c_2 \vee \bar{c}_1\bar{c}_2$

LNodeDisjoint = True iff $c_1\bar{c}_2 \vee \bar{c}_1 c_2 \vee \bar{c}_1\bar{c}_2$

CPT Construction – 4

CPT for Concept Nodes – 1

- The remaining issue is to construct CPTs for the concept nodes in X_C so that
  - $P(X_C \mid \tau)$, the joint probability distribution of all concept nodes in the subspace of τ, is consistent with all the given prior and conditional probabilities attached to the nodes in X_C
    - $P(C)$ or $P(C|O_C)$, where $O_C \subseteq \pi_C$, $\pi_C \neq \emptyset$, $O_C \neq \emptyset$
- Difficulties
  - CPTs are for the general space, but the given probabilities are for the subspace τ
  - Given probabilities may not be in the form of CPTs
  - Direct application of IPFP on joint probability $P(X_C \mid \tau)$ is expensive

CPT Construction – 5

CPT for Concept Nodes – 2

Initial Distribution:

$$Q_{(0)} = P_{init}(X) = \prod_{X_i \in X} P_{init}(X_i \mid \pi_i)$$

SD-IPFP for One Constraint:

$$Q_{(k)}(C_i|\pi_{C_i}) = Q_{(k-1)}(C_i|\pi_{C_i}) \cdot \frac{R(C_i|O_{C_i})}{Q_{(k-1)}(C_i|O_{C_i},\tau)} \cdot \alpha_{(k-1)}(\pi_{C_i})$$

Normalization Factor:

$$\alpha_{k-1}(\pi_{C_i}) = \frac{1}{\sum_{C_i \in \{c_i, \bar{c}_i\}} \frac{Q_{(k-1)}(C_i \mid \pi_{C_i}) \cdot R(C_i \mid O_{C_i})}{Q_{(k-1)}(C_i \mid O_{C_i}, \tau)}}$$

CPT Construction – 6

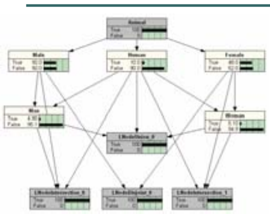
X_C = {Animal, Male, Female, Human, Man, Woman}

X_L = {LNodeUnion_0, LNodeIntersection_0, LNodeIntersection_1, LNodeDisjoint_0}

- *"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"; and*
- *"Human" is the* **union** *of "Man" and "Woman".*

- $P(Animal) = 0.5$
- $P(Male|Animal) = 0.5$
- $P(Female|Animal) = 0.48$
- $P(Human|Animal) = 0.1$
- $P(Man|Human) = 0.49$
- $P(Woman|Human) = 0.51$

CPT Construction – 7



DAG

CPT

Animal	
True	False
0.92752	0.07248

Animal	Human	
	True	False
True	0.18773	0.81227
False	0.0	1.0

Animal	Female	
	True	False
True	0.95469	0.04531
False	0.0	1.0

Animal	Male	
	True	False
True	0.95677	0.04323
False	0.0	1.0

Male	Human	Man	
		True	False
True	True	0.47049	0.52951
True	False	0.0	1.0
False	True	0.0	1.0
False	False	0.0	1.0

Female	Human	Woman	
		True	False
True	True	0.51433	0.48567
True	False	0.0	1.0
False	True	0.0	1.0
False	False	0.0	1.0

Semantics of *BayesOWL* – 1

The translated BN will be associated with a JPD

$$P'(X_C) = P(X_C|\tau)$$

on top of the standard DL semantics.

A description logic interpretation $\boldsymbol{I} = (\Delta^{\boldsymbol{I}}, .^{\boldsymbol{I}})$ consists of a non-empty domain of objects $\Delta^{\boldsymbol{I}}$ and an interpretation function $.^{\boldsymbol{I}}$. This function maps every concept to a subset of $\Delta^{\boldsymbol{I}}$, every role and attribute to a subset of $\Delta^{\boldsymbol{I}} \times \Delta^{\boldsymbol{I}}$, and every individual to an object of $\Delta^{\boldsymbol{I}}$. An interpretation $\boldsymbol{I}$ is a model for a concept C if $C^{\boldsymbol{I}}$ is non-empty, and C is said "satisfiable". Besides this description logic interpretation $\boldsymbol{I} = (\Delta^{\boldsymbol{I}}, .^{\boldsymbol{I}})$, in ***BayesOWL*** semantics, there is a function Pr to map each object $o \in \Delta^{\boldsymbol{I}}$ to a value between 0 and 1, $0 \leq \Pr(o) \leq 1$, and $\sum \Pr(o) = 1$, for all $o \in \Delta^{\boldsymbol{I}}$. This is the probability distribution over all the domain objects. For a class C: $P'(C) = \sum \Pr(o)$ for all $o \in C$. If C and D are classes and $C \subseteq D$, then $P'(C) \leq P'(D)$. Then, for a node C_i in X_C, $P'(C_i) = P(C_i|\boldsymbol{\tau})$ represents the probability distribution of an arbitrary object belonging (and not belonging) to the concept represented by C_i.

Semantics of *BayesOWL* – 2

In the translated BN, when all the L-Nodes are set to "True", all the logical relations specified in the original OWL ontology will be held, which means:

1. if B is a subclass of A then "$P'(b|\bar{a}) = 0 \wedge P'(a|b) = 1$";
2. if B is disjoint with A then "$P'(b|a) = 0 \wedge P'(a|b) = 0$";
3. if A is equivalent with B then "$P'(a) = P'(b)$";
4. if A is complement of B then "$P'(a) = 1 - P'(b)$";
5. if C is the intersection of C_1 and C_2 then "$P'(c|c_1, c_2) = 1 \wedge P'(c|\bar{c}_1) = 0 \wedge P'(c|\bar{c}_2) = 0 \wedge P'(c_1|c) = 1 \wedge P'(c_2|c) = 1$"; and
6. if C is the union of C_1 and C_2 then "$P'(c|\bar{c}_1, \bar{c}_2) = 0 \wedge P'(c|c_1) = 1 \wedge P'(c|c_2) = 1 \wedge P'(c_1|\bar{c}) = 0 \wedge P'(c_2|\bar{c}) = 0$".

Note it would be trivial to extend 5 and 6 to general case.

Semantics of *BayesOWL* – 3

Due to d-separation in the BN structure, additional conditional independencies may be imposed on the concept nodes in $\boldsymbol{X_C}$ in the translated BN.

1. serial connection: consider A is a parent superclass of B, B is a parent superclass of C, then the probability of an object o belonging to A and belonging to C is independent if o is known to be in B;
2. diverging connection: A is the parent superclass for both B and C, then B and C is conditionally independent given A;
3. converging connection: both B and C are parent superclasses of A, then B and C are assumed to be independent if nothing about A is known.

Reasoning – 1

Concept Satisfiability: $P(e|\boldsymbol{\tau}) = 0$**?**

Concept Overlapping: $P(e|C, \boldsymbol{\tau})$**=?**

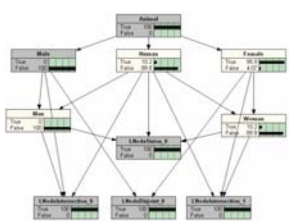
Concept Subsumption: Instead of find the most specific subsumer ***C*** of ***e***, find the one that is most similar to ***e*** based on "Jaccard Coefficient", i.e.

$$MSC(e, C) = P(e \cap C|\boldsymbol{\tau})/P(e \cup C|\boldsymbol{\tau})$$
$$= P(e \cap C|\boldsymbol{\tau})/(P(e|\boldsymbol{\tau}) + P(C|\boldsymbol{\tau}) - P(e \cap C|\boldsymbol{\tau}))$$

1. When only considering subsumers of ***e*** (i.e., ***P*(c|e,τ)=1**), ***MSC*** is reduced to ***P*(e|*C*,τ)**, and the ***C*** with the greatest ***MSC*** value is a most specific subsumer of ***e***.
2. The input description ***e*** can be uncertain, i.e., it is not restricted to hard evidences.

Reasoning – 2

$e = \neg Male \sqcap Animal$



- $MSC(e, Animal) = 0.5004$
- $MSC(e, Male) = 0.0$
- $MSC(e, Female) = \mathbf{0.9593}$
- $MSC(e, Human) = 0.0928$
- $MSC(e, Man) = 0.0$
- $MSC(e, Woman) = 0.1019$

Reasoning – 3

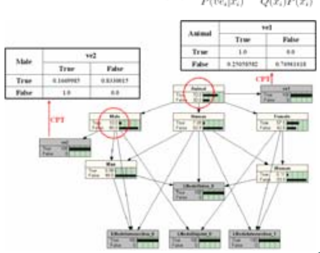
e': $P(Male) = 0.1$ and $P(Animal) = 0.7$

- $P(ve_1|Animal = True) = 1.0$
- $P(ve_1|Animal = False) = 0.25058582$
- $P(ve_2|Male = True) = 0.1669985$
- $P(ve_2|Male = False) = 1.0$

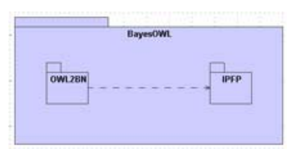
handle the soft evidence by Pearl's virtual evidence method

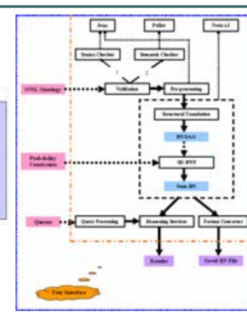
Likelihood Ratio: $L(X_i) = \frac{P(ve_i|x_i)}{P(ve_i|\overline{x_i})} = \frac{P(x_i)Q(\overline{x_i})}{Q(x_i)P(\overline{x_i})}$

- $MSC(e', Animal) = 0.4668$
- $MSC(e', Male) = 0.0667$
- $MSC(e', Female) = \mathbf{0.5753}$
- $MSC(e', Human) = 0.0676$
- $MSC(e', Man) = 0.0094$
- $MSC(e', Woman) = 0.0611$



The *OWL2BN* API





55/65

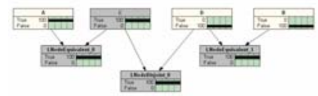
Comparison to Existing Works – 1

- The works closest to ***BayesOWL*** are P-CLASSIC and PTDL
  - Neither P-CLASSIC nor PTDL provides a method to construct CPTs. In contrast, one of ***BayesOWL***'s major contributions is its SD-IPFP mechanism to construct CPTs from given piece-wise probabilistic constraints.
  - Moreover, in ***BayesOWL***, by using L-Nodes, the "rdfs:subClassOf" relations (or the subsumption hierarchy) are separated from other logical relations, so the in-arcs to a concept node ***C*** will only come from its parent superclass nodes, which makes ***C***'s CPT smaller and easier to construct than P-CLASSIC or PTDL, especially in a domain with rich logical relations (it might be impossible for a domain expert to assign CPTs for some nodes, using P-CLASSIC or PTDL).
  - ***BayesOWL*** is specifically designed for OWL-DL taxonomy constructs.
  - ***BayesOWL*** is also engineering-oriented.
- Holi and Hyvönen's work for modeling an RDF(S) concept subsumption hierarchy
  - the arcs in the translated BN by are pointed from from child subconcept nodes to parent superconcept nodes
  - only deals with the "rdfs:subClassOf" relation
  - creates one node for each concept overlap, which generates a large DAG as well as hard to specify CPTs

56/65

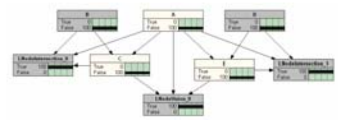
Comparison to Existing Works – 2

Example 1



Example 2

$$\left.\begin{array}{l} C \equiv A \sqcap B \\ E \equiv A \sqcap D \\ A \equiv C \sqcup E \end{array}\right\} \Rightarrow A \equiv (A \sqcap B) \sqcup (A \sqcap D) \Rightarrow A \equiv A \sqcap (B \sqcup D) \Leftrightarrow A \sqsubseteq (B \sqcup D)$$



57/65

Comparison to Existing Works – 3

- Compared to PR-OWL and OWL_QM
  - ***BayesOWL*** concerns explicitly more about the set or class memberships and logical relations rather than relationship among attributes.
  - ***BayesOWL*** is based on the model-theoretic semantics of OWL, so the modeling of uncertainty is in the granularity of concept classes.
  - Same as P-CLASSIC and PTDL, ***BayesOWL*** uses the standard BN model, while PR-OWL and OWL_QM do not.
  - ***BayesOWL*** trades the expressiveness with the simplicity.
- Some advantages of ***BayesOWL***
  - It translates a given ontology to a BN in a systematic and practical way and then treats ontological reasoning as probabilistic inferences in the translated BNs.
  - It is non-intrusive in the sense that neither OWL nor ontologies defined in OWL need to be modified.
  - It is flexible that one can translate either the entire ontology or part of it into BN depending on the needs.
  - It does not require availability of complete conditional probability distributions for CPTs, pieces of probabilistic information can be incorporated into the translated BN in a consistent fashion using SD-IPFP.
  - The cost of the approach is low and the burden to the user is minimal.
  - It can be easily extended to handle other ontology representation formalisms (syntax is not important, semantic matters), if not using OWL.

58/65

Summary

- What have already been done
  - proposed a method to encode probabilistic constraints for ontology classes and relations in OWL
  - defined a set of rules for translating an OWL taxonomy ontology into a BN DAG
  - provided a new algorithm SD-IPFP for efficient construction of CPTs
    - the translated BN is semantically consistent with the original ontology and satisfies all given probabilistic constraints
    - reasoning can be conducted as probabilistic inferences with potentially better, more accurate results
  - implemented the *OWL2BN* API based on the current settings
- What else could be done
  - extending the translation to include properties, instances and datatypes
  - supporting ontology mapping based on ***BayesOWL*** (Rong Pan's Ph.D. dissertation)
  - learning probabilities from existing web data (Yang Yu's M.S. thesis)

59/65

Conclusion and Future Research – 1

Major Contributions

- provides a non-intrusive and flexible method to translate an OWL taxonomy ontology into a BN
- provides a systematic and disciplined approach for CPT construction of the translated BN, and its extensions to solve more general BN modification problems
- proposes a method to markup probabilistic information using OWL statements
- implemented APIs to be used by other researchers and practitioners in ontology engineering

this research is the first to model uncertainty in OWL with a principled yet practical manner in the semantic web community

besides the theoretical foundation, its value is also reflected by its careful considerations in engineering which make ***BayesOWL*** easy to use by ontology designers and users.

60/65

Conclusion and Future Research – 2

Future – 1: Dealing with Properties and Instances

- Can we just adopt methods from P-CLASSIC directly?
  - needs a complicated probability computation mechanism across the set of p-classes
- Can we translate the whole ontology into one single standard Bayesian network?
  - the difficulty comes from the fact that a single concept ***C*** may be associated with more than one probability spaces when ***C*** acts in different roles
  - how to connect these two spaces is the hinge to completely resolve this issue
- Can we borrow methods from PRM or DAPER to build a BN based on instances?

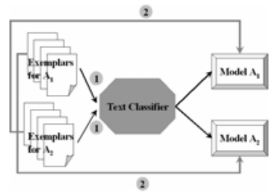
Conclusion and Future Research – 3

Future – 2: Dealing with Inconsistent Probabilistic Constraints

- the impact of the input constraint set's quality on the quality of solution
  - Existence: Under what condition will the input constraint set specify a multivariate joint distribution?
  - Uniqueness: Assume such joint distribution exists, when will it be unique?
  - Inconsistency: How to deal with inconsistent input constraint set which either oscillates in cycles or fails to converge
    - method 1: define some "distance measure" from the marginals of a distribution in the oscillating sequence to the given constraints, and choose the one that minimizes the distance measure as the best solution
    - method 2: "missing data methods", which aims to modify the given inconsistent probabilistic constraints to make it workable by analyzing and processing the inconsistent data (assume data exists) using methods from Statistics

Conclusion and Future Research – 4

Future – 3: Learning Probabilities from Web Data



Cross-Classification using Rainbow

Initial solution: using text classification technique by explicitly associating a concept with exemplars retrieved and selected automatically from WWW.

How to obtain high quality exemplars (both positive and negative) automatically?

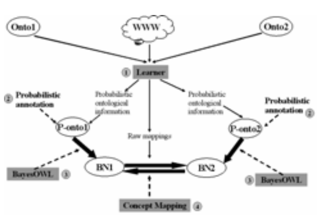
(1) how relevant to the concept?
(2) how comprehensive in capturing all important aspects of the concept?

(1) how to form the query strings based on given ontologies?
(2) how to pre-process the documents crawled back?

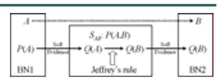
issue: the semantics of a concept in a document represents more close to its meaning in natural language, while the semantics of a concept in OWL is based on model-theoretics.

Conclusion and Future Research – 5

Future – 4: Supporting Ontology Mapping

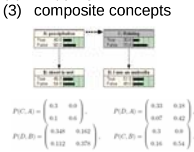


Proposed Ontology Mapping Framework



Mapping Concept *A* in Ontology1 to *B* in Ontology2

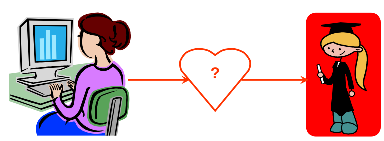
Remaining Issues:
(1) from 1:1 to 1:n, n:1, n:m
(2) mapping reduction
(3) composite concepts



Conclusion and Future Research – 6

Future – 5: Other Comments

- design and develop GUI interfaces for the current prototype implementation
- develop an ontology for standard Bayesian networks
- develop an ontology of probability theory



Acknowledgement

- Academic Support
  - Dr. Yun Peng
  - Dr. Timothy Finin
  - Dr. Anupam Joshi
  - Dr. Konstantinos Kalpakis
  - Dr. Tim Oates
  - Dr. Lina Zhou
  - CS faculty and staff
- eBiquity lab mates
- Family members and friends