# Foreword

This volume contains the papers presented at the *5th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2009)*, held as a part of the *8th International Semantic Web Conference (ISWC 2009)* at the Westfields Conference Center near Washington, DC, USA, October 26, 2009. It contains 6 technical papers and 3 position papers, which were selected in a rigorous reviewing process, where each paper was reviewed by at least four program committee members.

The International Semantic Web Conference is a major international forum for presenting visionary research on all aspects of the Semantic Web. The International Workshop on Uncertainty Reasoning for the Semantic Web is an exciting opportunity for collaboration and cross-fertilization between the uncertainty reasoning community and the Semantic Web community. Effective methods for reasoning under uncertainty are vital for realizing many aspects of the Semantic Web vision, but the ability of current-generation Web technology to handle uncertainty is extremely limited. Recently, there has been a groundswell of demand for uncertainty reasoning technology among Semantic Web researchers and developers. This surge of interest creates a unique opening to bring together two communities with a clear commonality of interest but little history of interaction. By capitalizing on this opportunity, URSW could spark dramatic progress toward realizing the Semantic Web vision.

**Audience:** The intended audience for this workshop includes the following: (1) researchers in uncertainty reasoning technologies with interest in Semantic Web and Web-related technologies; (2) Semantic Web developers and researchers; (3) people in the knowledge representation community with interest in the Semantic Web; (4) ontology researchers and ontological engineers; (5) Web services researchers and developers with interest in the Semantic Web; and (6) developers of tools designed to support Semantic Web implementation, e.g., Jena, Protégé, and Protégé-OWL developers.

**Topics:** We intended to have an open discussion on any topic relevant to the general subject of uncertainty in the Semantic Web (including fuzzy theory, probability theory, and other approaches). Therefore, the following list should be just an initial guide: (1) syntax and semantics for extensions to Semantic Web languages to enable representation of uncertainty; (2) logical formalisms to support uncertainty in Semantic Web languages; (3) probability theory as a means of assessing the likelihood that terms in different ontologies refer to the same or similar concepts; (4) architectures for applying plausible reasoning to the problem of ontology mapping; (5) using fuzzy approaches to deal with imprecise concepts within ontologies; (6) the concept of a probabilistic ontology and its relevance to the Semantic Web; (7) best practices for representing uncertain, incomplete, ambiguous, or controversial information in the Semantic Web; (8) the role of uncertainty as it relates to Web services; (9) interface protocols with support for uncertainty as a means to improve interoperability among Web services; (10) uncertainty reasoning techniques applied to trust issues in the Semantic Web; (11) existing implementations of uncertainty reasoning tools in the context of the Semantic Web; (12) issues and techniques for integrating tools for representing and reasoning with uncertainty; and (13) the future of uncertainty reasoning for the Semantic Web.

We wish to thank all authors who submitted papers and all workshop participants for fruitful discussions. We would like to thank the program committee members and external referees for their timely expertise in carefully reviewing the submissions.

October 2009

Fernando Bobillo
Paulo C. G. da Costa
Claudia d'Amato
Nicola Fanizzi
Kathryn B. Laskey
Kenneth J. Laskey
Thomas Lukasiewicz
Trevor Martin
Matthias Nickles
Michael Pool
Pavel Smrž

# Workshop Organization

## Program Chairs

Fernando Bobillo (University of Zaragoza, Spain)
Paulo C. G. da Costa (George Mason University, USA)
Claudia d'Amato (University of Bari, Italy)
Nicola Fanizzi (University of Bari, Italy)
Kathryn B. Laskey (George Mason University, USA)
Kenneth J. Laskey (MITRE Corporation, USA)
Thomas Lukasiewicz (University of Oxford, UK)
Trevor Martin (University of Bristol, UK)
Matthias Nickles (University of Bath, UK)
Michael Pool (Convera, Inc., USA)
Pavel Smrž (Brno University of Technology, Czech Republic)

## Program Committee

Fernando Bobillo (University of Zaragoza, Spain)
Silvia Calegari (University of Milano-Bicocca, Italy)
Rommel Carvalho (George Mason University, USA)
Paulo C. G. da Costa (George Mason University, USA)
Fabio Gagliardi Cozman (University of São Paulo, Brazil)
Claudia d'Amato (University of Bari, Italy)
Nicola Fanizzi (University of Bari, Italy)
Marcelo Ladeira (University of Brasilia, Brazil)
Kathryn B. Laskey (George Mason University, USA)
Kenneth J. Laskey (MITRE Corporation, USA)
Thomas Lukasiewicz (University of Oxford, UK)
Anders L. Madsen (Hugin Expert A/S, Denmark)
Trevor Martin (University of Bristol, UK)
Matthias Nickles (University of Bath, UK)
Jeff Z. Pan (University of Aberdeen, UK)
Yun Peng (University of Maryland, Baltimore County, USA)
Michael Pool (Convera, Inc., USA)
Livia Predoiu (University of Mannheim, Germany)
Guilin Qi (University of Karlsruhe, Germany)
Carlos Henrique Ribeiro (Instituto Tecnológico de Aeronáutica, Brazil)
David Robertson (University of Edinburgh, UK)
Daniel Sánchez (University of Granada, Spain)
Sergej Sizov (University of Koblenz-Landau, Germany)
Pavel Smrž (Brno University of Technology, Czech Republic)
Giorgos Stoilos (National Technical University of Athens, Greece)
Umberto Straccia (ISTI-CNR, Italy)

Andreas Tolk (Old Dominion University, USA)
Johanna Völker (University of Karlsruhe, Germany)
Peter Vojtáš (Charles University, Czech Republic)

## External Reviewers

Zhiqiang Gao

# Table of Contents

## Technical Papers

## Position Papers

# Technical Papers

# Probabilistic Ontology and Knowledge Fusion for Procurement Fraud Detection in Brazil

Rommel N. Carvalho[1], Kathryn B. Laskey[1], Paulo C. G. Costa[1], Marcelo Ladeira[2], Laécio L. Santos[2], and Shou Matsumoto[2],

[1] George Mason University
4400 University Drive
Fairfax, VA 22030-4400 USA
rommel.carvalho@gmail.com, {klaskey, pcosta}@gmu.edu
[2] University of Brasilia
Campus Universitário Darcy Ribeiro
Brasilia – DF 70910-900 Brazil
mladeira@unb.br, {laecio, cardialfly}@gmail.com

**Abstract.** To cope with society's demand for transparency and corruption prevention, the Brazilian Office of the Comptroller General (CGU) has carried out a number of actions, including: awareness campaigns aimed at the private sector; campaigns to educate the public; research initiatives; and regular inspections and audits of municipalities and states. Although CGU has collected information from hundreds of different sources - Revenue Agency, Federal Police, and others - the process of fusing all this data has not been efficient enough to meet the needs of CGU's decision makers. Therefore, it is natural to change the focus from data fusion to knowledge fusion. As a consequence, traditional syntactic methods must be augmented with techniques that represent and reason with the semantics of databases. However, commonly used approaches fail to deal with uncertainty, a dominant characteristic in corruption prevention. This paper presents the use of Probabilistic OWL (PR-OWL) to design and test a model that performs information fusion to detect possible frauds in procurements involving Federal money. To design this model, a recently developed tool for creating PR-OWL ontologies was used with support from PR-OWL specialists and careful guidance from a fraud detection specialist from CGU.

**Keywords:** Probabilistic Ontology, PR-OWL, Ontology, Procurement Fraud Detection, Knowledge Fusion, MEBN, UnBBayes.

## 1 Introduction

A primary responsibility of the Brazilian Office of the Comptroller General (CGU) is to prevent and detect government corruption. To carry out this mission, CGU must gather information from a variety of sources and combine it to evaluate whether further action, such as an investigation, is required. One of the most difficult challenges is the information explosion. Auditors must fuse vast quantities of information from a variety of sources in a way that highlights its relevance to decision makers and helps them focus their efforts on the most critical cases. This is no trivial duty. The Growing Acceleration Program (PAC) alone has a budget greater than 250 billion dollars with more than one thousand projects only on the state of Sao Paulo (http://www.brasil.gov.br/pac/). All of these have to be audited and inspected by CGU – and, in spite having only three thousand employees. Therefore, CGU must optimize its processes in order to carry out its mission.

The Semantic Web (SW), like the document web that preceded it, is based on radical notions of information sharing. These ideas [1] include: (i) the Anyone can say Anything about Any topic (AAA) slogan; (ii) the open world assumption, in which we assume there is always more information that could be known, and (iii) nonunique naming, which appreciates the reality that different speakers on the Web might use different names to define the same entity. In a fundamental departure from assumptions of traditional information systems architectures, the Semantic Web is intended to provide an environment in which information sharing can thrive and a network effect of knowledge synergy is possible. But this style of information gathering can generate a chaotic landscape rife with confusion, disagreement and conflict.

We call an environment characterized by the above assumptions a Radical Information Sharing (RIS) environment. The challenge facing SW architects is therefore to avoid the natural chaos to which RIS environments are prone, and move to a state characterized by information sharing, cooperation and collaboration. According to [1], one solution to this challenge lies in modeling, and this is where ontologies languages like Web Ontology Language (OWL) come in.

As it will be shown in Section 3, the domain of procurement fraud detection is a RIS environment. However, uncertainty is ubiquitous to knowledge fusion. Uncertainty is especially important to applications such as fraud detection, in which perpetrators seek to conceal illicit intentions and activities, making crisp assertions extremely hard and rare. In such environments, partial (not complete) or approximate (not exact) information is more the rule than the exception.

Bayesian networks (BNs) have been widely applied to draw inferences to information and knowledge fusion in the presence of uncertainty. However, according to [2] BNs are not expressive enough for many real-world applications. More specifically, BNs assume a simple attribute-value representation – that is, each problem instance involves reasoning about the same fixed number of attributes, with only the evidence values changing from problem instance to problem instance. Complex problems on the scale of the semantic web often involve intricate relationships among many variables, and the limited representational power of BNs is insufficient for building useful, detailed models.

Multi-Entity Bayesian Network (MEBN) logic can represent and reason with uncertainty about any propositions that can be expressed in first-order logic [3]. Probabilistic OWL (PR-OWL) uses MEBN's strengths to provide a framework for building probabilistic ontologies (PO), a major step towards semantically aware, probabilistic knowledge fusion systems [4]. This paper uses PR-OWL to design and test a model for fusing information to detect possible frauds in procurements involving Federal funds.

The paper is organized as follows. Section 2 introduces Multi-Entity Bayesian Networks (MEBN), an expressive Bayesian logic, and PR-OWL, an extension of the OWL language that can represent probabilistic ontologies having MEBN as its underlying logic. Section 3 presents a case study from CGU to demonstrate the power of PR-OWL ontologies for knowledge representation and fusion. Finally, Section 4 presents some concluding remarks.

## 2 MEBN and PR-OWL

Multi-Entity Bayesian Networks (MEBN) [5 and 6] extend BNs (BN) to achieve first-order expressive power. MEBN represents knowledge as a collection of MEBN Fragments (MFrags), which are organized into MEBN Theories (MTheories).

An MFrag contains random variables (RVs) and a fragment graph representing dependencies among these RVs. An MFrag is a template for a fragment of a Bayesian network. It is instantiated by binding its arguments to domain entity identifiers to create instances of its RVs. There are three kinds of RV: context, resident and input. Context RVs represent conditions that must be satisfied for the distributions represented in the MFrag to apply. Input nodes represent RVs that may influence the distributions defined in the MFrag, but whose distributions are defined in other MFrags. Distributions for resident RV instances are defined in the MFrag. Distributions for resident RVs are defined by specifying local distributions conditioned on the values of the instances of their parents in the fragment graph.

A set of MFrags represents a joint distribution over instances of its random variables. MEBN provides a compact way to represent repeated structure in a BN. An important advantage of MEBN is that there is no fixed limit on the number of RV instances, and the random variable instances are dynamically instantiated as needed.

An MTheory is a set of MFrags that satisfies conditions of consistency ensuring the existence of a unique joint probability distribution over its random variable instances.

To apply an MTheory to reason about particular scenarios, one needs to provide the system with specific information about the individual entity instances involved in the scenario. On receipt of this information, Bayesian inference can be used both to answer specific questions of interest (e.g., how likely is it that a particular procurement is being directed to a specific enterprise?) and to refine the MTheory (e.g., each new tactical situation includes additional statistical data about the likelihood of a given attack for that set of circumstances). Bayesian inference is used to perform both problem specific inference and learning in a sound, logically coherent manner (for more details see [6 and 7]).

State-of-the-art systems are increasingly adopting ontologies as a means to ensure formal semantic support for knowledge sharing [8, 9, 10, 11, 12, and 13]. Representing and reasoning with uncertainty is becoming recognized as an essential capability in many domains. A common error is to provide support for uncertainty representation by just annotating ontologies with numerical probabilities. This approach leads to brittleness, as too much information is lost due to the lack of a representational scheme that can capture structural nuances of the probabilistic information. More expressive representation formalisms are needed [4].

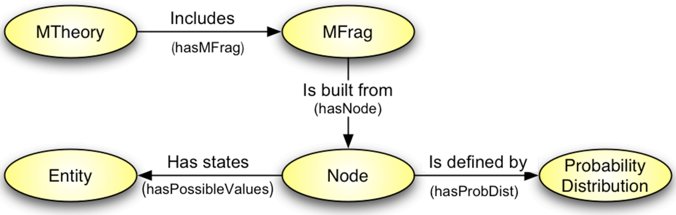


**Fig. 1.** PR-OWL main concepts.

Probabilistic Ontologies (PR-OWL) [14 and 15] was proposed as a more expressive formalism for representing knowledge in domains characterized by uncertainty. Figure 1 presents the main concepts needed to define an MTheory in PR-OWL. In the diagram, the ellipses represent the general classes, while the arcs represent the main relationships among the classes.

The procurement fraud detection probabilistic ontology was built in UnBBayes-MEBN, a tool for building and reasoning with PR-OWL probabilistic ontologies. UnBBayes-MEBN was the first software to implement PR-OWL/MEBN (see [16, 17, 18, 19] for more details). UnBBayes-MEBN supports Multi-Entity Bayesian Network (MEBN) and enables creation and editing of Probabilistic Ontologies in PR-OWL [18]. The MEBN/PR-OWL Graphical User Interface (GUI) [16] allows users to define MFrags and make probabilistic queries. UnBBayes-MEBN also implements an algorithm for generating a Situation Specific Bayesian Network (SSBN) [18, 19], which is an ordinary BN created by instantiating instances of the MFrags to respond to a probabilistic query. Once the SSBN is generated, the inference engine (Reasoning) is called to process findings and update beliefs. UnBBayes-MEBN uses the Protégé-OWL library to load and save PR-OWL files (IO) in a format compatible with OWL. It supports first order logic context node evaluation (FOL), through the use of the PowerLoom library. It also defines and implements a built-in mechanism for typing and recursion. Finally, it permits the definition of dynamic conditional probabilistic tables.

UnBBayes has proven to be a simple, yet powerful, tool for designing probabilistic ontologies and for uncertain reasoning in complex situations such as procurement fraud detection. It is straightforward to use and provides powerful features (e.g. dynamic table) not available in systems (e.g., Quiddity) previously employed to reason with PR-OWL/MEBN knowledge bases.

## 3 Procurement Fraud Detection

A major source of corruption is the procurement process. Although laws attempt to ensure a competitive and fair process, perpetrators find ways to turn the process to their advantage while appearing to be legitimate. This is why a specialist has didactically structured the different kinds of procurement frauds CGU has dealt with in past years.

These different fraud types are characterized by criteria, such as business owners who work as a front for the company, use of accounting indices that are not common practice, etc. Indicators have been established to help identify cases of each of these fraud types. For instance, one principle that must be followed in public procurement is that of competition. Every public procurement should establish minimum requisites necessary to guarantee the execution of the contract in order to maximize the number of participating bidders. Nevertheless, it is common to have a fake competition when different bidders are, in fact, owned by the same person. This is usually done by having someone as a front for the enterprise, which is often someone with little or no education.

The ultimate goal of this case study is to structure the specialist knowledge in a way that an automated system can reason with the evidence in a manner similar to the specialist. Such an automated system is intended to support specialists and to help train new specialists, but not to replace them. Initially, a few simple criteria were selected as a proof of concept. Nevertheless, it is shown that the model can be incrementally updated to incorporate new criteria. In this process, it becomes clear that a number of different sources must be consulted to come up with the necessary indicators to create new and useful knowledge for decision makers about the procurements.

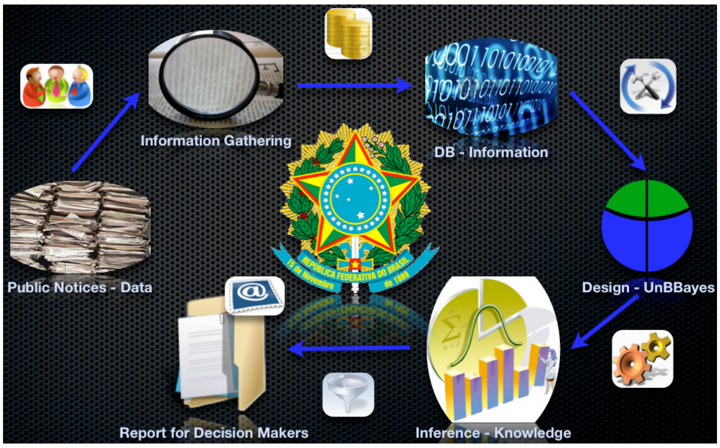


**Fig. 2.** Procurement fraud detection overview.

Figure 2 presents an overview of the procurement fraud detection process. The data for our case study represent several requests for proposal and auctions that are issued by the Federal, State and Municipal Offices (Public Notices – Data). As the focus of the work is in representing the specialist knowledge and reasoning through probabilistic ontologies and not in the collection of information, the idea is that the analysts that work at CGU, already making audits and inspections, accomplish the collection of information through questionnaires that can specifically be created for the collecting of indicators for the selected criteria (Information Gathering). These questionnaires can be created using a system that is already in production at CGU. Once they are answered the necessary information is going to be available (DB – Information). Hence, UnBBayes, using the probabilistic ontology designed by experts (Design – UnBBayes), will be able to collect these millions of items of information and transform them into dozens or hundreds of items of knowledge, through logic and probabilistic inference, e.g. procurement announcements, contracts, reports, etc - a huge amount of data - are analyzed allowing the gathering of relevant relations and properties - a large amount of information - which in turn are used to draw some conclusions about possible irregularities - a smaller number of items of knowledge (Inference – Knowledge). This knowledge can be filtered so that only the procurements that show a probability higher than a threshold, e.g. 20%, are automatically forwarded to the responsible department along with the inferences about potential fraud and the supporting evidence (Report for Decision Makers).

The criteria selected by the specialist were the use of accounting indices and the demand of experience in just one contract. There are four common types of indices that are usually used as requirements in procurements (ILC, ILG, ISG, and IE). Any other type could indicate a made-up index specifically designed to direct the procurement to some specific company. The greater the numbers of uncommon accounting indices used by the procurement the more suspicious it is, i.e. the higher the chance of having fraud. In addition, a procurement specifies a minimum value for these accounting indices. The minimum value that is usually required is 1.0. The higher this minimum value, the more the competition is narrowed, and therefore the higher the chance the procurement is being directed to some company.

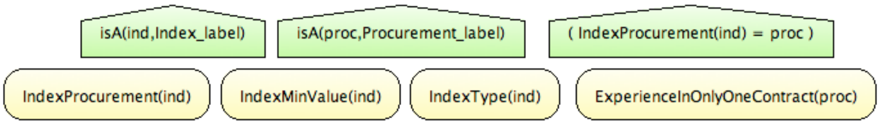

isA(ind,Index_label)
isA(proc,Procurement_label)
( IndexProcurement(ind) = proc )
IndexProcurement(ind)
IndexMinValue(ind)
IndexType(ind)
ExperienceInOnlyOneContract(proc)

**Fig. 3.** ProcurementRequirement MFrag.

The other criterion, demanding proof of experience in only one contract, is suspect because in almost every case, the experience is not gained only by a particular contract, but also by doing it over and over again in different contracts. It does not matter if you have built 1,000 ft2 of wall in just one contract or 100 ft2 in 10 different contracts. The experience gained will be basically the same.

The procurement fraud detection model was developed as a probabilistic ontology (using PR-OWL) to define its semantics and uncertain characteristics. The MTheory created for the model, using UnBBayes-MEBN, was divided into three different MFrags.

The first, Figure 3, presents the criteria required from a company to participate in the procurement, containing information about the type of accounting index (ILC, ILG, ISG, IE, and Other) and the minimum value for it (between 0 and 1, between 1 and 2, between 2 and 3, and greater than 3). This MFrag also contains information about where a specific index is used (which procurement), and if the procurement demands experience in only one contract.

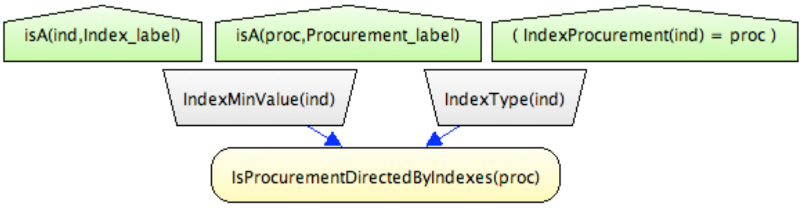

isA(ind,Index_label)
isA(proc,Procurement_label)
( IndexProcurement(ind) = proc )
IndexMinValue(ind)
IndexType(ind)
IsProcurementDirectedByIndexes(proc)

**Fig. 4.** DirectingProcurementByIndexes MFrag.

The second, Figure 4, represents whether procurement is being directed to a specific company by the use of unusual accounting indices. As explained before, this analysis is based on the type of the index and the minimum value it requires. This evaluation takes into consideration every index used in a specific procurement, hence it is dynamic.

The last MFrag, Figure 5, represents the overall possibility that procurement is being directed to a specific company based on the result of its being directed by the use of unusual indices and by the requirement of experience in only one contract, as explained before.

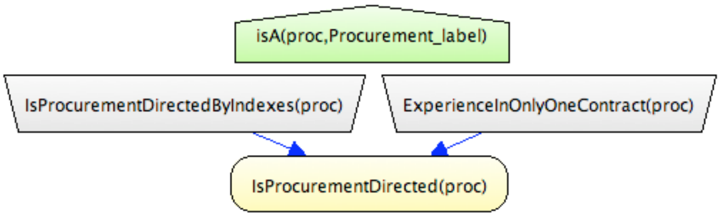

isA(proc,Procurement_label)
IsProcurementDirectedByIndexes(proc)
ExperienceInOnlyOneContract(proc)
IsProcurementDirected(proc)

**Fig. 5.** DirectingProcurement MFrag.

To test the model, two scenarios, that represent the two groups of suspect and non suspect procurements, were chosen from a set of real cases, as shown:

- Suspect procurement (*proc1*):
  - ind1 = ILC >= 2.0;
  - ind2 = ILG >= 1.5;
  - ind3 = Other >= 3.0.
  - It demands experience in only one contract.
- Non suspect procurement (*proc2*):
  - ind4 = IE >= 1.0;

  - ind5 = ILG >= 1.0;
  - ind6 = ILC >= 1.0;
  - It does not demand experience in only one contract.

The information above was introduced in our model as known entities and findings. After that we queried the system to give us information about the node *IsProcurementDirected(proc)* for both *proc1* and *proc2*. UnBBayes-MEBN than executed the SSBN algorithm and generated the same node structure as shown in Figure 6, because both procurements have three accounting indices and information about the demanding experience in only one contract. However, as expected, the parameters and findings are different giving different results to the query, as shown below:

- Non suspect procurement:
  - 0.01% that the procurement was directed to a specific company by using accounting indices;
  - 0.10% that the procurement was directed to a specific company.
- Suspect procurement:
  - 55.00% that the procurement was directed to a specific company by using accounting indices;
  - 29.77%, when the information about demanding experience in only one contract was omitted, and 72.00%, when it was given, that the procurement was directed to a specific company.

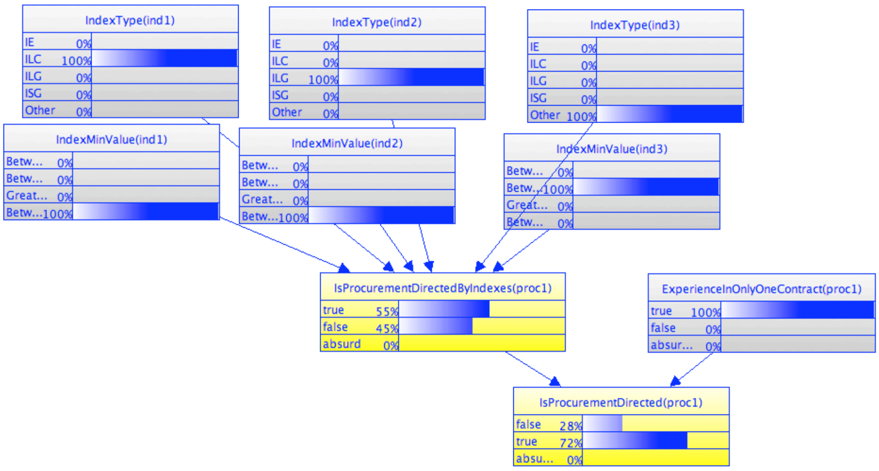


**Fig. 6.** Generated SSBN for query IsProcurementDirected(proc1).

The specialist analyzed and agreed with the knowledge generated by the probabilistic ontology reasoned developed using PR-OWL/MEBN in UnBBayes. He stated that the probabilities represent, semantically (i.e. high, medium, and low chance), what he would think when analyzing the same entities and findings.

Although the SSBNs generated for this proof of concept present the same structure, it is common to have a different one as the context varies from procurement to procurement. For instance, we have come across several procurements that have all four common indices and some other different ones. In this case, if there are two additional indices (*ind5* and *ind6*), then the resulting SSBN would have two more copies for nodes *IndexType(index)* and*IndexMinValue(index)*. This would make the use of BN not applicable. The ability to make multiple copies of nodes based on a context is only available in a more expressive formalism, as MEBN.

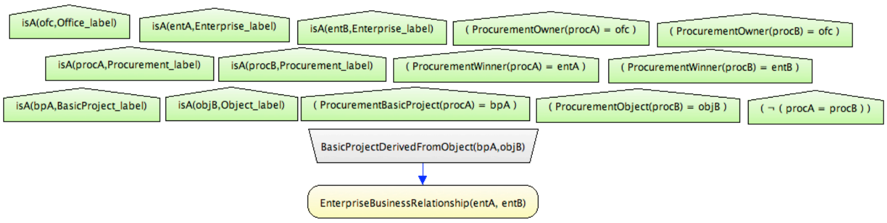

isA(ofc,Office_label)
isA(entA,Enterprise_label)
isA(entB,Enterprise_label)
( ProcurementOwner(procA) = ofc )
( ProcurementOwner(procB) = ofc )
isA(procA,Procurement_label)
isA(procB,Procurement_label)
( ProcurementWinner(procA) = entA )
( ProcurementWinner(procB) = entB )
isA(bpA,BasicProject_label)
isA(objB,Object_label)
( ProcurementBasicProject(procA) = bpA )
( ProcurementObject(procB) = objB )
( ¬ ( procA = procB ) )
BasicProjectDerivedFromObject(bpA,objB)
EnterpriseBusinessRelationship(entA, entB)

**Fig. 7.** EnterpriseBusinessNetwork MFrag.

An additional capability not available with BN is to specify constraints on applicability of knowledge. Such constraints can only be implemented in a more expressive language. As we are dealing with BN formalism it is only natural to think of a formalism that extends BN. MEBN, as a Bayesian first-order logic, makes it possible to define these constraints using FOL.

Figure 7 presents the constraints (context nodes) necessary to model the fraud detection scenarios considered here. In this MFrag, the criterion is to identify if there is a suspicious business relationship between enterprises *entA* and *entB*. The more cases where enterprise B wins a procurement that the basic project was developed by enterprise A, the higher the chance they have some kind of personal business relationship, which means that it is more likely that enterprise B is developing the basic projects in such a way that will favor enterprise A, inhibiting the desired competition.

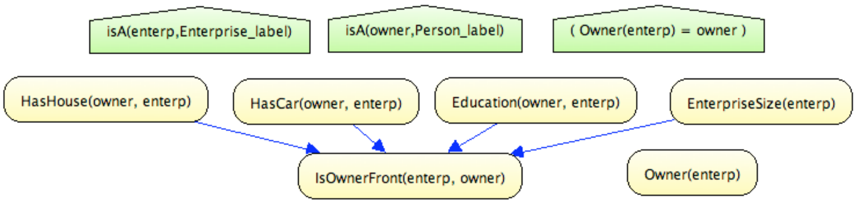


**Fig. 8.** OwnerFront MFrag.

Since the designed model is restricted to just two criteria, the team started to think about other criteria that could be incorporated and tested further. Figure 8 presents the suggested MFrag for detecting owners that act as a front to the real owner of the company (the person who really has the power to make decisions and that gets all the money), by looking up their socio-economic attributes and checking the size of the company. In other words, if a company is highly profitable, yet has an owner with little education, low income, no car, no house, etc, then the company is probably a front.

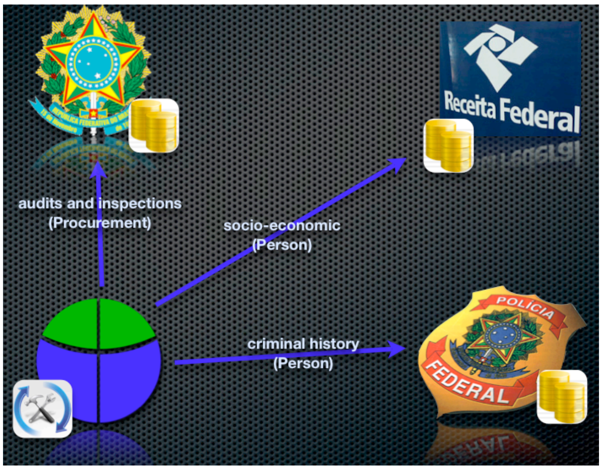


**Fig. 9.** Knowledge fusion from different Government Offices DBs.

From the criteria presented and modeled in this Section, we can clearly see the need for a principled way of dealing with uncertainty. But what is the role of Semantic Web in this domain? Well, it is easy to see that our domain of fraud detection is a RIS environment. The data CGU has available does not come only from its audits and inspections. In fact, much complementary information can be retrieved from other Federal Agencies, including Federal Revenue Agency, Federal Police, and others. Imagine we have information about the enterprise that won the procurement, and we want to know information about its owners, such as their personal data and annual income. This type of information is not available at CGU's Data Base (DB), but must be retrieved from the Federal Revenue Agency's DB. Once the information about the owners is available, it might be useful to check their criminal history. For that (see Figure 9), information from the Federal Police must be used. In this example, we have different sources saying different things about the same person: thus, the AAA slogan applies. Moreover, there might be other Agencies with crucial information related to our person of interest; in other words, we are operating in an open world. Finally, to make this sharing and integration process possible, we have to make sure we are talking about the same person, who may (especially in case of fraud) be known by different names in different contexts.

## 5 Conclusion

The problem that CGU and many other Agencies have faced of processing all the available data into useful knowledge is starting to be solved with the use of probabilistic ontologies, as the procurement fraud detection model showed. Besides fusing the information available, the designed model was able to represent the specialist knowledge for the two real cases we evaluated. UnBBayes reasoning given the evidence and using the designed model were accurate both in suspicious and non suspicious scenarios. These results are encouraging, suggesting that a fuller development of our proof of concept system is promising.

In addition, it is fairly easy to introduce new criteria and indicators in the model in an incremental way. Thus, new rules for identifying fraud can be added without rework. After a new rule is incorporated into the model, a set of new tests can be added to the previous one with the objective of always validating the new model proposed, without doing everything from scratch.

Furthermore, the use of this formalism through UnBBayes allows advantages such as impartiality in the judgment of irregularities in procurements (given the same conditions the system will always deliver the same result), scalability (capacity to analyze thousands of procurements in a short time when compared to human capacity) and a joint analysis of large volumes of indicators (the higher the number of indicators to examine jointly the more difficult it is for the specialist analysis to be objective and consistent).

As a next step, CGU is choosing new criteria to be incorporated into the designed probabilistic ontology. This next set of criteria will require information from different Brazilian Agencies' databases. Therefore, the semantic power of ontologies with the uncertainty handling capability of PR-OWL will be extremely useful for fusing information from multiple databases.

**Acknowledgments.** Rommel Carvalho gratefully acknowledges full support from the Brazilian Office of the Comptroller General (CGU) for the research reported in this paper, and its employees involved in this research, especially Mário Vinícius Claussen Spinelli, the domain expert.

## References

1. Allemang, D. & Hendler, J. A. 2008. Semantic web for the working ontologist modeling in RDF, RDFS and OWL. Elsevier, ISBN 978-0-12-373556-0, United States.
2. Costa, P. C. G., Laskey, K. B., Takikawa, M., Pool, M., Fung, F., and Wright, E. J. 2005. MEBN logic: A Key Enabler for Network Centric Warfare. *In Proceedings of the 10th International Command and Control Research and Technology Symposium (10th ICCRTS).* McLean, Virginia, USA, CCRP publications.
3. Laskey, K. B., Mahoney, S. M., and Wright, E. 2001. Hypothesis Management in Situation-Specific Network Construction. *Uncertainty in Artificial Intelligence: Proceedings of the Seventeenth Conference*, San Mateo, CA, Morgan Kaufman.
4. Laskey, K. B., Costa, P. C. G., and Janssen, T. 2008. Probabilistic Ontologies for Knowledge Fusion. In *Proceedings of the 11th International Conference on Information Fusion.*
5. Laskey, K. B. and Costa, P. .C. G. 2005. Of Klingons and Starships: Bayesian Logic for the 23rd Century. *In Uncertainty in Artificial Intelligence: Proceedings of the Twenty-first Conference (UAI 2005).* AUAI Press: Edinburgh, Scotland.
6. Laskey, K. B. 2008. MEBN: A language for first-order Bayesian knowledge bases. *In Artificial Intelligence*, Volume 172, Issues 2-3, February 2008, Pages 140-178

7. Mahoney, S. and Laskey, K. B. 1998. Constructing Situation Specific Belief Networks, In Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98). San Francisco, CA: Morgan Kaufmann.
8. Chen, H., and Wu, Z.. 2003. On Case-Based Knowledge Sharing in Semantic Web. *In Tools with Artificial Intelligence*, IEEE International Conference on, 0:200. Vol. 0. Los Alamitos, CA, USA, IEEE Computer Society.
9. Chen, H., Wu, Z., and Xu, J. 2003. KB-Grid: Enabling Knowledge Sharing on the Semantic Web. *In Challenges of Large Applications in Distributed Environments*, International Workshop on, 0:70. Vol. 0. Los Alamitos, CA, USA: IEEE Computer Society.
10. Costa, Paulo C. G., Chang, KC., Laskey, K. B., and Carvalho, R. N. 2009. A Multi-Disciplinary Approach to High Level Fusion in Predictive Situational Awareness. *In Proceedings of the 12th International Conference on Information Fusion*. Seattle, WA, USA.
11. Dadzie, AS., Bhagdev, R., Chakravarthy, A., Chapman, S., Iria, J., Lanfranchi, V., Magalhães, J., Petrelli, D., and Ciravegna. F. 2008. Applying semantic web technologies to knowledge sharing in aerospace engineering. *Journal of Intelligent Manufacturing* 20, no. 5 (6): 611-623. doi:10.1007/s10845-008-0141-1.
12. Kings, N. J. and Davies, J. 2009. Semantic Web for Knowledge Sharing. *In Semantic Knowledge Management*, 103-111. http://dx.doi.org/10.1007/978-3-540-88845-1_8.
13. Veres, G. V., Huynh, T. D., Nixon, M. S., Smart, P. R. and Shadbolt, N. R. 2006. The Military Knowledge Information Fusion Via Semantic Web Technologies. http://eprints.ecs.soton.ac.uk/14278/.
14. Costa, P. C. G. 2005. Bayesian Semantics for the Semantic Web. PhD Diss. Department of Systems Engineering and Operations Research, George Mason University. 315p, Fairfax, VA, USA.
15. Costa, P. C. G., Laskey, K. B., and Laskey, K. J. 2005. PR-OWL: A Bayesian Ontology Language for the Semantic Web. In *Proceedings of the ISWC Workshop on Uncertainty Reasoning for the Semantic Web*, Galway, Ireland.
16. Carvalho, R. N., Ladeira, M., Santos, L. L., and Costa, P. C. 2007. A GUI Tool for Plausible Reasoning in the Semantic Web using MEBN. In *Proceedings of the Seventh international Conference on intelligent Systems Design and Applications,* 381-386. ISDA. IEEE Computer Society, Washington, DC, USA.
17. Carvalho, R. N., Ladeira, M., Santos, L. L., Matsumoto, S., and Costa, P. C. G. 2008. UnBBayes-MEBN: Comments on Implementing a Probabilistic Ontology Tool. In *Proceedings of the IADIS International Conference on Applied Computing,* 211-218.
18. Costa, P. C. G., Ladeira, M, Carvalho, R. N., Santos, L. L., Matsumoto, S., and Laskey, K. B. 2008. A First-Order Bayesian Tool for Probabilistic Ontologies. In *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference,* 631-636. Menlo Park, California, USA, The AAAI Press.
19. Carvalho, R. N., Ladeira, M., Santos, L. L., Matsumoto, S., and Costa, P. C. G. 2009. A GUI Tool for Plausible Reasoning in the Semantic Web Using MEBN. In *Book Innovative Applications in Data Mining,* 17-45. DOI: 10.1007/978-3-540-88045-5_2. Springer Berlin / Heidelberg.

# Combining Semantic Web Search with the Power of Inductive Reasoning

Claudia d'Amato[1], Nicola Fanizzi[1], Bettina Fazzinga[2],
Georg Gottlob[3,4], and Thomas Lukasiewicz[3,5]

[1] Dipartimento di Informatica, Università degli Studi di Bari, Italy
{claudia.damato,fanizzi}@di.uniba.it
[2] Dipartimento di Elettronica, Informatica e Sistemistica, Università della Calabria, Italy
bfazzinga@deis.unical.it
[3] Computing Laboratory, University of Oxford, UK
{georg.gottlob,thomas.lukasiewicz}@comlab.ox.ac.uk
[4] Oxford-Man Institute of Quantitative Finance, University of Oxford, UK
[5] Institut für Informationssysteme, TU Wien, Austria

**Abstract.** Extensive research activities are recently directed towards the Semantic Web as a future form of the Web. Consequently, Web search as the key technology of the Web is evolving towards some novel form of Semantic Web search. A very promising recent approach to such Semantic Web search is based on combining standard Web search with ontological background knowledge and using standard Web search engines as the main inference motor of Semantic Web search. In this paper, we propose to further enhance this approach to Semantic Web search by the use of inductive reasoning techniques. This adds especially the important ability to handle inconsistencies, noise, and incompleteness, which are very likely to occur in distributed and heterogeneous environments, such as the Web. We report on a prototype implementation of the new approach and experimental results.

## 1 Introduction

Web search [3] as the key technology of the Web is about to change radically with the development of the *Semantic Web* [2]. As a consequence, the elaboration of a new search technology for the Semantic Web, called *Semantic Web search* [6], is currently an extremely hot topic, both in Web-related companies and in academic research. In particular, there is a fast growing number of commercial and academic Semantic Web search engines. The research can be roughly divided into two main directions. The first (and most common) one is to develop a new form of search for searching the pieces of data and knowledge that are encoded in the new representation formalisms of the Semantic Web (e.g., [6]), while the second (and less explored) direction is to use the data and knowledge of the Semantic Web in order to add some semantics to Web search (e.g., [9]).

A very promising recent representative of the second direction to Semantic Web search has been presented in [8]. The approach is based on (i) using ontological (unions of) conjunctive queries (which may contain negated subqueries) as Semantic Web search queries, (ii) combining standard Web search with ontological background knowledge, (iii) using the power of Semantic Web formalisms and technologies, and (iv) using standard Web search engines as the main inference motor of Semantic Web search. It consists of an offline ontology compilation step, based on deductive reasoning techniques, and an online query processing step. In this paper, we propose to further enhance this approach to Semantic Web search by the use of inductive reasoning techniques for the offline ontology compilation step. To our knowledge, this is the first combination of Semantic Web search with inductive reasoning. The paper's main contributions can be summarized as follows:

- We develop a combination of Semantic Web search as presented in [8] with an inductive reasoning technique (based on similarity search [11] for retrieving the resources that likely belong to a query concept [5]). The latter serves in an offline ontology compilation step to compute completed semantic annotations.
- Importantly, the new approach to Semantic Web search can handle inconsistencies, noise, and incompleteness in Semantic Web knowledge bases, which are all very likely to occur in distributed and heterogeneous environments, such as the Web. We provide several examples illustrating this important advantage of the new approach.
- We report on a prototype implementation of the new approach in the context of desktop search. We also provide very positive experimental results for the precision and the recall of the new approach, comparing it to the deductive approach in [8].

## 2 System Overview

The overall architecture of our Semantic Web search system is shown in Fig. 1. It consists of the *Interface*, the *Query Evaluator*, and the *Inference Engine* (Fig. 1, dark parts), where the Query Evaluator is implemented on top of standard Web *Search Engines*. Standard *Web* pages and their objects are enriched by *Annotation* pages, based on an *Ontology*.

We thus assume that there are semantic annotations to standard Web pages and to objects on standard Web pages. Note that such annotations are starting to be widely available for a large class of Web resources, especially with the Web 2.0. Semantic annotations about Web pages and objects may also be automatically learned from the Web pages and the objects to be annotated (see, e.g., [4]), and/or they may be extracted from existing ontological knowledge bases on the Semantic Web. Another important standard assumption that we make is that Web pages and their objects have unique identifiers.

For example, in a very simple scenario, a Web page $i_1$ may contain information about a Ph.D. student $i_2$, called Mary, and two of her papers, namely, a conference paper $i_3$ entitled "*Semantic Web search*" and a journal paper $i_4$ entitled "*Semantic Web search engines*" and published in 2008. A simple HTML page representing this scenario is shown in Fig. 2, left side. There may now exist one semantic annotation each for the Web page, the Ph.D. student Mary, the journal paper, and the conference paper. The annotation for the Web page may simply encode that it mentions Mary and the two papers, while the one for Mary may encode that she is a Ph.D. student with the name Mary and the author of the papers $i_3$ and $i_4$. The annotation for the paper $i_3$ may encode that $i_3$ is a conference paper and has the title "*Semantic Web search*", while the one for the paper $i_4$ may encode that $i_4$ is a journal paper, authored by Mary, has the title "*Semantic Web search engines*", was published in 2008, and has the keyword "RDF". The semantic annotations of $i_1$, $i_2$, $i_3$, and $i_4$ are formally expressed as the sets of axioms $\mathcal{A}_{i_1}$, $\mathcal{A}_{i_2}$, $\mathcal{A}_{i_3}$, and $\mathcal{A}_{i_4}$, respectively:

$$\begin{aligned}
\mathcal{A}_{i_1} &= \{\mathit{contains}(i_1, i_2), \mathit{contains}(i_1, i_3), \mathit{contains}(i_1, i_4)\},\\
\mathcal{A}_{i_2} &= \{\mathit{PhDStudent}(i_2),\ \mathit{name}(i_2, \text{“\textit{mary}”}),\ \mathit{isAuthorOf}(i_2, i_3),\ \mathit{isAuthorOf}(i_2, i_4)\},\\
\mathcal{A}_{i_3} &= \{\mathit{ConferencePaper}(i_3), \mathit{title}(i_3, \text{“\textit{Semantic Web search}”})\},\\
\mathcal{A}_{i_4} &= \{\mathit{JournalPaper}(i_4),\ \mathit{hasAuthor}(i_4, i_2),\ \mathit{title}(i_4, \text{“\textit{Semantic Web search engines}”}),\\
&\qquad \mathit{yearOfPublication}(i_4, 2008), \mathit{keyword}(i_4, \text{“RDF”})\}.
\end{aligned} \tag{1}$$

**Inference Engine.** Using an ontology containing some background knowledge, these semantic annotations are then further enhanced in an offline ontology compilation step, where the *Inference Engine* adds all properties that can be deduced from the semantic

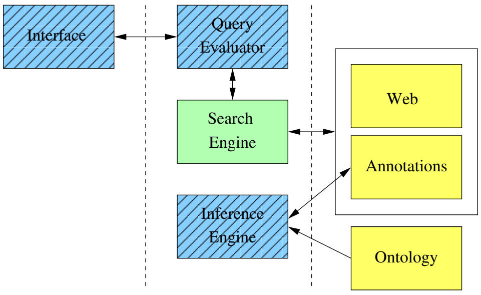
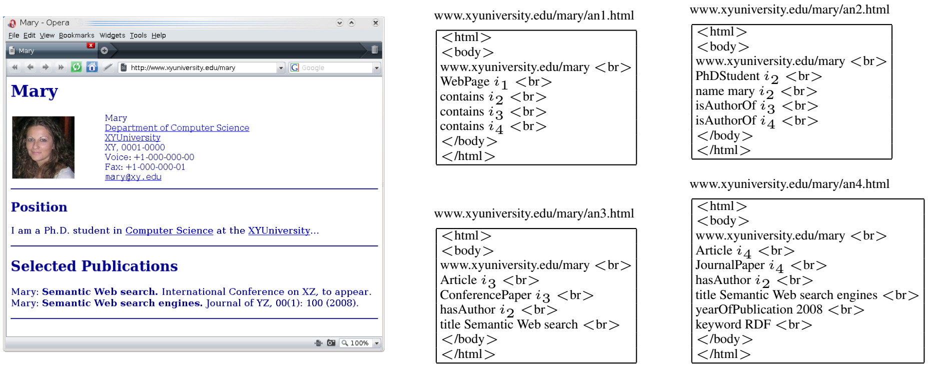


**Fig. 1.** System architecture.


Mary - Opera
File Edit View Bookmarks Widgets Tools Help
Mary
http://www.xyuniversity.edu/mary
Google
100%

**Mary**

Mary
Department of Computer Science
XYUniversity
XY, 0001-0000
Voice: +1-000-000-00
Fax: +1-000-000-01
mary@xy.edu

**Position**

I am a Ph.D. student in Computer Science at the XYUniversity...

**Selected Publications**

Mary: **Semantic Web search.** International Conference on XZ, to appear.
Mary: **Semantic Web search engines.** Journal of YZ, 00(1): 100 (2008).

www.xyuniversity.edu/mary/an1.html

```
<html>
<body>
www.xyuniversity.edu/mary <br>
WebPage i1 <br>
contains i2 <br>
contains i3 <br>
contains i4 <br>
</body>
</html>
```

www.xyuniversity.edu/mary/an2.html

```
<html>
<body>
www.xyuniversity.edu/mary <br>
PhDStudent i2 <br>
name mary i2 <br>
isAuthorOf i3 <br>
isAuthorOf i4 <br>
</body>
</html>
```

www.xyuniversity.edu/mary/an3.html

```
<html>
<body>
www.xyuniversity.edu/mary <br>
Article i3 <br>
ConferencePaper i3 <br>
hasAuthor i2 <br>
title Semantic Web search <br>
</body>
</html>
```

www.xyuniversity.edu/mary/an4.html

```
<html>
<body>
www.xyuniversity.edu/mary <br>
Article i4 <br>
JournalPaper i4 <br>
hasAuthor i2 <br>
title Semantic Web search engines <br>
yearOfPublication 2008 <br>
keyword RDF <br>
</body>
</html>
```

**Fig. 2.** Left side: HTML page $p$; right side: four HTML pages $p_1$, $p_2$, $p_3$, and $p_4$, which encode (completed) semantic annotations for $p$ and the objects on $p$.

annotations and the ontology. In [8], we assume a deductive such step, while here we propose and explore an inductive one. The resulting (*completed*) semantic annotations are then published as Web pages, so that they can be searched by standard Web search engines. For example, an ontology may contain the knowledge that (i) conference and journal papers are articles, (ii) conference papers are not journal papers, (iii) *isAuthorOf* relates scientists and articles, (iv) *isAuthorOf* is the inverse of *hasAuthor*, and (v) *hasFirstAuthor* is a functional binary relationship, which is formally expressed by:

$$\begin{array}{l} \mathit{ConferencePaper} \sqsubseteq \mathit{Article},\ \mathit{JournalPaper} \sqsubseteq \mathit{Article},\ \mathit{ConferencePaper} \sqsubseteq \neg \mathit{JournalPaper},\\ \exists \mathit{isAuthorOf} \sqsubseteq \mathit{Scientist},\ \exists \mathit{isAuthorOf}^- \sqsubseteq \mathit{Article},\ \mathit{isAuthorOf}^- \sqsubseteq \mathit{hasAuthor},\\ \mathit{hasAuthor}^- \sqsubseteq \mathit{isAuthorOf},\ (\mathsf{funct}\ \mathit{hasFirstAuthor}). \end{array} \tag{2}$$

Using this ontological knowledge, we can derive from the above annotations that the two papers $i_3$ and $i_4$ are also articles, and both authored by John. These resulting searchable (completed) semantic annotations of (objects on) standard Web pages are published as HTML Web pages with pointers to the respective object pages, so that they (in addition to the standard Web pages) can be searched by standard search engines. For example, the HTML pages for the completed semantic annotations of the above $\mathcal{A}_{i_1}$, $\mathcal{A}_{i_2}$, $\mathcal{A}_{i_3}$, and $\mathcal{A}_{i_4}$ are shown in Fig. 2, right side. Note that on the HTML page of each individual, its identifier is located beside the atomic concept below the row specifying the URIs. Practically, such an identifier may simply be the HTML address of the Web page/object's annotation page. For example, considering the HTML pages of Fig. 2, the individual described by $p_4$ is $i_4$, and the one described by $p_2$ is $i_2$. Observe that we use a plain textual representation of the completed semantic annotations in order to allow their processing by existing standard search engines for the Web. It is important to point out that this textual representation is simply a list of properties, each eventually along with an identifier or a data value as attribute value, and it can thus immediately be encoded as a list of RDF triples.

**Query Evaluator.** The *Query Evaluator* (see Fig. 1) reduces each Semantic Web search query of the user in an online query processing step to a sequence of standard Web search queries on standard Web and annotation pages, which are then processed by a standard Web *Search Engine*. The Query Evaluator also collects the results and re-transforms them into a single answer which is returned to the user. As an example of a Semantic Web search query, one may ask for all Ph.D. students who have published an article in 2008 with RDF as a keyword, which is formally expressed as follows:

$$\begin{aligned} Q(x) = \exists y\, (&\mathit{PhDStudent}(x) \wedge \mathit{isAuthorOf}(x,y) \wedge \mathit{Article}(y) \wedge \\ &\mathit{yearOfPublication}(y, 2008) \wedge \mathit{keyword}(y, \text{“}RDF\text{”}))\,. \end{aligned}$$

This query is transformed into the two queries $Q_1 = \mathit{PhDStudent}$ AND *isAuthorOf* and $Q_2 = \mathit{Article}$ AND *“yearOfPublication* 2008” AND *“keyword* RDF”, which can both be submitted to a standard Web search engine, such as Google. The result of the original query $Q$ is then built from the results of the two queries $Q_1$ and $Q_2$. Note that a graphical user interface, such as the one of Google's advanced search, or even a natural language interface can help to hide the conceptual complexity of ontological queries to the user.

## 3 Semantic Web Search

We now introduce Semantic Web knowledge bases and the syntax and semantics of Semantic Web search queries to such knowledge bases. We then generalize the PageRank technique to our approach. We assume the reader is familiar with the syntax and the semantics of Description Logics (DLs) [1], which we use as underlying ontology languages.

**Semantic Web Knowledge Bases.** Intuitively, a Semantic Web knowledge base consists of a background TBox and a collection of ABoxes, one for every concrete Web page and for every object on a Web page. For example, the homepage of a scientist may be such a concrete Web page and be associated with an ABox, while the publications on the homepage may be such objects, which are also associated with one ABox each.

We assume pairwise disjoint sets $\mathbf{D}$, $\mathbf{A}$, $\mathbf{R}_A$, $\mathbf{R}_D$, $\mathbf{I}$, and $\mathbf{V}$ of atomic datatypes, atomic concepts, atomic roles, atomic attributes, individuals, and data values, respectively. Let $\mathbf{I}$ be the disjoint union of two sets $\mathbf{P}$ and $\mathbf{O}$ of *Web pages* and *Web objects*, respectively. Informally, every $p \in \mathbf{P}$ is an identifier for a concrete Web page, while every $o \in \mathbf{O}$ is an identifier for a concrete object on a concrete Web page. We assume the atomic roles *links_to* between Web pages and *contains* between Web pages and Web objects. The former represents the link structure between concrete Web pages, while the latter encodes the occurrences of concrete Web objects on concrete Web pages.

**Definition 1.** A *semantic annotation* $\mathcal{A}_a$ for a Web page or object $a \in \mathbf{P} \cup \mathbf{O}$ is a finite set of concept membership axioms $A(a)$, role membership axioms $P(a,b)$, and attribute membership axioms $U(a,v)$, where $A \in \mathbf{A}$, $P \in \mathbf{R}_A$, $U \in \mathbf{R}_D$, $b \in \mathbf{I}$, and $v \in \mathbf{V}$. A *Semantic Web knowledge base* $KB = (\mathcal{T}, (\mathcal{A}_a)_{a \in \mathbf{P} \cup \mathbf{O}})$ consists of a TBox $\mathcal{T}$ and one semantic annotation $\mathcal{A}_a$ for every Web page and object $a \in \mathbf{P} \cup \mathbf{O}$.

Informally, a Semantic Web knowledge base consists of some background terminological knowledge and some assertional knowledge for every concrete Web page and for every concrete object on a Web page. The background terminological knowledge may be an ontology from some global Semantic Web repository or an ontology defined locally by the user site. In contrast to the background terminological knowledge, the assertional knowledge will be directly stored on the Web (on annotation pages like the described standard Web pages) and is thus accessible via Web search engines.

*Example 1. (Scientific Database).* We use a DL knowledge base $KB = (\mathcal{T}, \mathcal{A})$ to specify some simple information about scientists and their publications. The sets of atomic concepts, atomic roles, atomic attributes, and data values are:

$$\begin{aligned}&\mathbf{A} = \{\mathit{Scientist}, \mathit{Article}, \mathit{ConferencePaper}, \mathit{JournalPaper}\},\\&\mathbf{R}_A = \{\mathit{hasAuthor}, \mathit{isAuthorOf}, \mathit{contains}\},\ \mathbf{R}_D = \{\mathit{name}, \mathit{title}, \mathit{yearOfPublication}\},\\&\mathbf{V} = \{\text{“}\mathit{mary}\text{”}, \text{“}\mathit{Semantic\ Web\ search}\text{”}, 2008, \text{“}\mathit{Semantic\ Web\ search\ engines}\text{”}\}.\end{aligned}$$

Let $\mathbf{I} = \mathbf{P} \cup \mathbf{O}$ be the set of individuals, where $\mathbf{P} = \{i_1\}$ is the set of Web pages, and $\mathbf{O} = \{i_2, i_3, i_4\}$ is the set of Web objects on the Web page $i_1$. The TBox $\mathcal{T}$ contains the axioms in Eq. 2. Then, a Semantic Web knowledge base is given by $KB = (\mathcal{T}, (\mathcal{A}_a)_{a \in \mathbf{P} \cup \mathbf{O}})$, where the semantic annotations of the individuals in $\mathbf{P} \cup \mathbf{O}$ are the ones in Eq. 1.

**Semantic Web Search Queries.** We use unions of conjunctive queries with negated conjunctive subqueries as Semantic Web search queries to Semantic Web knowledge bases. We now first define the syntax of Semantic Web search queries and then the semantics of positive and general such queries.

*Syntax.* Let $\mathbf{X}$ be a finite set of variables. A *term* is either a Web page $p \in \mathbf{P}$, a Web object $o \in \mathbf{O}$, a data value $v \in \mathbf{V}$, or a variable $x \in \mathbf{X}$. An *atomic formula* (or *atom*) $\alpha$ is of one of the following forms: (i) $d(t)$, where $d$ is an atomic datatype, and $t$ is a term; (ii) $A(t)$, where $A$ is an atomic concept, and $t$ is a term; (iii) $P(t, t')$, where $P$ is an atomic role, and $t, t'$ are terms; and (iv) $U(t, t')$, where $U$ is an atomic attribute, and $t, t'$ are terms. An *equality* has the form $=(t, t')$, where $t$ and $t'$ are terms. A *conjunctive formula* $\exists \mathbf{y}\, \phi(\mathbf{x}, \mathbf{y})$ is an existentially quantified conjunction of atoms $\alpha$ and equalities $=(t, t')$, which have free variables among $\mathbf{x}$ and $\mathbf{y}$.

**Definition 2.** A *Semantic Web search query* $Q(\mathbf{x})$ is an expression $\bigvee_{i=1}^{n} \exists \mathbf{y}_i\, \phi_i(\mathbf{x}, \mathbf{y}_i)$, where each $\phi_i$ with $i \in \{1, \ldots, n\}$ is a conjunction of atoms $\alpha$ (also called *positive atoms*), negated conjunctive formulas $\mathit{not}\, \psi$, and equalities $=(t, t')$, which have free variables among $\mathbf{x}$ and $\mathbf{y}_i$, and the $\mathbf{x}$'s are exactly the free variables of $\bigvee_{i=1}^{n} \exists \mathbf{y}_i\, \phi_i(\mathbf{x}, \mathbf{y}_i)$.

Intuitively, Semantic Web search queries are unions of conjunctive queries, which may contain negated conjunctive queries in addition to atoms and equalities as conjuncts.

*Example 2. (Scientific Database cont'd).* Two Semantic Web search queries are:

$$\begin{aligned}Q_1(x) = {} & (\mathit{Scientist}(x) \wedge \mathit{not\ doctoralDegree}(x, \text{“}\mathit{oxford\ university}\text{”}) \wedge \mathit{worksFor}(x,\\& \text{“}\mathit{oxford\ university}\text{”})) \vee (\mathit{Scientist}(x) \wedge \mathit{doctoralDegree}(x, \text{“}\mathit{oxford\ university}\text{”}) \wedge\\& \mathit{not\ worksFor}(x, \text{“}\mathit{oxford\ university}\text{”}));\\Q_2(x) = {} & \exists y\, (\mathit{Scientist}(x) \wedge \mathit{worksFor}(x, \text{“}\mathit{oxford\ university}\text{”}) \wedge \mathit{isAuthorOf}(x, y) \wedge\\& \mathit{not\ ConferencePaper}(y) \wedge \mathit{not}\ \exists z\, \mathit{yearOfPublication}(y, z)).\end{aligned}$$

Informally, $Q_1(x)$ asks for scientists who are either working for *oxford university* and did not receive their Ph.D. from that university, or who received their Ph.D. from *oxford university* but do not work for it. Whereas query $Q_2(x)$ asks for scientists of *oxford university* who are authors of at least one unpublished non-conference paper. Note that when searching for scientists, the system automatically searches for all subconcepts (known according to the background ontology), such as e.g. Ph.D. students or computer scientists.

**Semantics of Positive Search Queries.** We now define the semantics of positive Semantic Web search queries, which are free of negations, in terms of ground substitutions via the notion of logical consequence.

A search query $Q(\mathbf{x})$ is *positive* iff it contains no negated conjunctive subqueries. A *(variable) substitution* $\theta$ maps variables from $\mathbf{X}$ to terms. A substitution $\theta$ is *ground* iff it maps to Web pages $p \in \mathbf{P}$, Web objects $o \in \mathbf{O}$, and data values $v \in \mathbf{V}$. A closed first-order formula $\phi$ is a *logical consequence* of a knowledge base $KB = (\mathcal{T}, (\mathcal{A}_a)_{a \in \mathbf{P} \cup \mathbf{O}})$, denoted $KB \models \phi$, iff every first-order model $\mathcal{I}$ of $\mathcal{T} \cup \bigcup_{a \in \mathbf{P} \cup \mathbf{O}} \mathcal{A}_a$ also satisfies $\phi$.

**Definition 3.** Given a Semantic Web knowledge base $KB$ and a positive Semantic Web search query $Q(\mathbf{x})$, an *answer* for $Q(\mathbf{x})$ to $KB$ is a ground substitution $\theta$ for the variables $\mathbf{x}$ (which are exactly the free variables of $Q(\mathbf{x})$) with $KB \models Q(\mathbf{x}\theta)$.

*Example 3. (Scientific Database cont'd).* Consider the Semantic Web knowledge base $KB$ of Example 1 and the following positive Semantic Web search query, asking for all scientists who author at least one published journal paper:

$$Q(x) = \exists y\, (\mathit{Scientist}(x) \wedge \mathit{isAuthorOf}(x, y) \wedge \mathit{JournalPaper}(y) \wedge \exists z\, \mathit{yearOfPublication}(y, z)).$$

An answer for $Q(x)$ to $KB$ is $\theta = \{x/i_2\}$. Recall that $i_2$ represents the scientist Mary.

**Semantics of General Search Queries.** We next define the semantics of general Semantic Web search queries by reduction to the semantics of positive ones, interpreting negated conjunctive subqueries $\mathit{not}\ \psi$ as the lack of evidence about the truth of $\psi$. That is, negations are interpreted by a closed-world semantics on top of the open-world semantics of DLs (we refer to [8] for more motivation and background).

**Definition 4.** Given a Semantic Web knowledge base $KB$ and search query

$$Q(\mathbf{x}) = \bigvee_{i=1}^{n} \exists \mathbf{y}_i\, \phi_{i,1}(\mathbf{x}, \mathbf{y}_i) \wedge \cdots \wedge \phi_{i,l_i}(\mathbf{x}, \mathbf{y}_i) \wedge \mathit{not}\ \phi_{i,l_i+1}(\mathbf{x}, \mathbf{y}_i) \wedge \cdots \wedge \mathit{not}\ \phi_{i,m_i}(\mathbf{x}, \mathbf{y}_i)\,,$$

an *answer* for $Q(\mathbf{x})$ to $KB$ is a ground substitution $\theta$ for the variables $\mathbf{x}$ such that $KB \models Q^+(\mathbf{x}\theta)$ and $KB \not\models Q^-(\mathbf{x}\theta)$, where $Q^+(\mathbf{x})$ and $Q^-(\mathbf{x})$ are defined as follows:

$$Q^+(\mathbf{x}) = \bigvee_{i=1}^{n} \exists \mathbf{y}_i\, \phi_{i,1}(\mathbf{x}, \mathbf{y}_i) \wedge \cdots \wedge \phi_{i,l_i}(\mathbf{x}, \mathbf{y}_i) \text{ and}$$
$$Q^-(\mathbf{x}) = \bigvee_{i=1}^{n} \exists \mathbf{y}_i\, \phi_{i,1}(\mathbf{x}, \mathbf{y}_i) \wedge \cdots \wedge \phi_{i,l_i}(\mathbf{x}, \mathbf{y}_i) \wedge (\phi_{i,l_i+1}(\mathbf{x}, \mathbf{y}_i) \vee \cdots \vee \phi_{i,m_i}(\mathbf{x}, \mathbf{y}_i))\,.$$

Roughly, a ground substitution $\theta$ is an answer for $Q(\mathbf{x})$ to $KB$ iff (i) $\theta$ is an answer for $Q^+(\mathbf{x})$ to $KB$, and (ii) $\theta$ is not an answer for $Q^-(\mathbf{x})$ to $KB$, where $Q^+(\mathbf{x})$ is the positive part of $Q(\mathbf{x})$, while $Q^-(\mathbf{x})$ is the positive part of $Q(\mathbf{x})$ combined with the complement of the negative one. Observe that both $Q^+(\mathbf{x})$ and $Q^-(\mathbf{x})$ are positive queries.

*Example 4. (Scientific Database cont'd).* Consider the Semantic Web knowledge base $KB = (\mathcal{T}, (\mathcal{A}_a)_{a \in \mathbf{P} \cup \mathbf{O}})$ of Example 1 and the following general Semantic Web search query, asking for Mary's unpublished non-journal papers:

$$Q(x) = \exists y\, (\mathit{Article}(x) \wedge \mathit{hasAuthor}(x, y) \wedge \mathit{name}(y, \text{“}\mathit{mary}\text{”}) \wedge \mathit{not}\ \mathit{JournalPaper}(x) \wedge$$
$$\mathit{not}\ \exists z\, \mathit{yearOfPublication}(x, z)).$$

An answer for $Q(x)$ to $KB$ is given by $\theta = \{\mathbf{x}/i_3\}$. Recall that $i_3$ represents an unpublished conference paper entitled "*Semantic Web search*". Observe that the membership axioms $\mathit{Article}(i_3)$ and $\mathit{hasAuthor}(i_2, i_3)$ do not appear in the semantic annotations $\mathcal{A}_a$ with $a \in \mathbf{P} \cup \mathbf{O}$, but they can be inferred from them using the background ontology $\mathcal{T}$.

**Ranking Answers.** As for the ranking of all answers for a Semantic Web search query $Q$ to a Semantic Web knowledge base $KB$ (i.e., ground substitutions for all free variables in $Q$, which correspond to tuples of Web pages, Web objects, and data values), we use a generalization of the PageRank technique: rather than considering only Web pages and the link structure between Web pages (expressed through the role *links_to* here), we also consider Web objects, which may occur on Web pages (expressed through the role *contains*), and which may also be related to other Web objects via other roles. More concretely, we define the *ObjectRank* of a Web page or an object $a$ as follows:

$$R(a) = d \cdot \textstyle\sum_{b \in B_a} R(b) \,/\, N_b + (1-d) \cdot E(a)\,,$$

where (i) $B_a$ is the set of all Web pages and Web objects that relate to $a$, (ii) $N_b$ is the number of Web pages and Web objects that relate from $b$, (iii) $d$ is a damping factor, and (iv) $E$ associates with every Web page and every Web object a source of rank.

## 4 Deductive Offline Ontology Compilation

In this section, we describe the (deductive) offline ontology reasoning step, which compiles the implicit terminological knowledge in the TBox of a Semantic Web knowledge base into explicit membership axioms in the ABox, i.e., in the semantic annotations of Web pages / objects, so that it (in addition to the standard Web pages) can be searched by standard Web search engines. For the online query processing step, see [8].

The compilation of TBox knowledge into ABox knowledge is formalized as follows. Given a satisfiable Semantic Web knowledge base $KB = (\mathcal{T}, (\mathcal{A}_a)_{a \in \mathbf{P} \cup \mathbf{O}})$, the *simple completion* of $KB$ is the Semantic Web knowledge base $KB' = (\emptyset, (\mathcal{A}_a{}')_{a \in \mathbf{P} \cup \mathbf{O}})$ such that every $\mathcal{A}_a{}'$ is the set of all concept memberships $A(a)$, role memberships $P(a,b)$, and attribute memberships $U(a,v)$ that logically follow from $\mathcal{T} \cup \bigcup_{a \in \mathbf{P} \cup \mathbf{O}} \mathcal{A}_a$, where $A \in \mathbf{A}$, $P \in \mathbf{R}_A$, $U \in \mathbf{R}_D$, $b \in \mathbf{I}$, and $v \in \mathbf{V}$. Informally, for every Web page and object, the simple completion collects all available and deducible facts (whose predicate symbols shall be usable in search queries) in a completed semantic annotation.

*Example 5.* Consider the TBox $\mathcal{T}$ of Example 1 and the semantic annotations $(\mathcal{A}_a)_{a \in \mathbf{P} \cup \mathbf{O}}$ of Example 1. The simple completion contains in particular the new axioms *Article*$(i_3)$, *hasAuthor*$(i_3, i_2)$, and *Article*$(i_4)$. The first two are added to $A_{i_3}$ and the last one to $A_{i_4}$.

As shown in [8], general quantifier-free search queries to a Semantic Web knowledge base $KB$ over *DL-Lite*$_{\mathcal{A}}$ [10] as underlying DL can be evaluated on the simple completion of $KB$ (which contains only compiled but no explicit TBox knowledge anymore). Similar results hold when the TBox of $KB$ is equivalent to a Datalog program, and the query is fully general. Hence, the simple completion assures (i) always a sound query processing and (ii) a complete query processing in many cases. For this reason, and since completeness of query processing is actually not that much an issue in the inherently incomplete Web, we propose to use the simple completion as the basis of our Semantic Web search.

Once the completed semantic annotations are computed, we encode them as HTML pages, so that they are searchable via standard keyword search. Specifically, we build one HTML page for the semantic annotation $\mathcal{A}_a$ of each individual $a \in \mathbf{P} \cup \mathbf{O}$. That is, for each individual $a$, we build a page $p$ containing all the atomic concepts whose argument is $a$ and all the atomic roles/attributes where the first argument is $a$ (see Section 2).

## 5 Inductive Offline Ontology Compilation

We now describe an inductive inference based on similarity search, which we propose to use instead of deductive inference for offline ontology compilation in our approach to Semantic Web search. Section 6 then summarizes the central advantages of this proposal.

**Inductive Inference Based on Similarity Search.** In *similarity search* [11], the basic idea is to find the most similar object(s) to a query object (i.e., the one to be classified) with respect to a similarity (or dissimilarity) measure. We review the basics of the $k$-nearest-neighbor ($k$-NN) method applied to the Semantic Web context [5]. The objective is to induce an approximation for a discrete-valued target hypothesis function $h\colon \mathit{IS} \rightarrow V$ from a space of instances *IS* to a set of values $V = \{v_1, \ldots, v_s\}$ standing for the classes (concepts) that have to be predicted. Let $x_q$ be the query instance whose class-membership is to be determined. Using a dissimilarity measure, the set of the $k$-nearest (pre-classified) training instances relative to $x_q$ is selected: $NN(x_q) = \{x_1, \ldots, x_k\}$. Hence, the $k$-NN algorithm approximates $h$ for classifying $x_q$ on the grounds of the value that $h$ is known to assume for the training instances in $NN(x_q)$. Precisely, the value is decided by means of a weighted majority voting procedure: it is the most *voted* value by the instances in $NN(x_q)$ weighted by the similarity of the neighbor individual. The estimate of the hypothesis function for the query individual is:

$$\hat{h}(x_q) := \operatorname*{argmax}_{v \in V} \sum_{i=1}^{k} w_i \delta(v, h(x_i)) \,, \tag{3}$$

where $\delta$ returns 1 in case of matching arguments and 0 otherwise, and, given a dissimilarity measure $d$, the weights $w_i$ are determined by $w_i = 1/d(x_i, x_q)$.

Observe that this setting assigns to the query instance $x_q$ a value, which stands for one in a set of pairwise disjoint concepts (corresponding to the value set $V$). In a multi-relational setting, as those of the Semantic Web (SW) context, this assumption cannot be made in general, since it is well known that an individual may be an instance of more than one concept. The problem is also related to the closed-world assumption (CWA) usually made in the knowledge discovery context. To deal with the open-world assumption (OWA), generally adopted for the SW representations, the absence of information on whether a training instance $x$ belongs to the extension of a query concept $Q$ should not be interpreted negatively, as in the standard settings which adopt the CWA, rather, it should count as neutral (uncertain) information. Assuming this alternate viewpoint, the multi-class classification problem is transformed into a ternary one and the $V = \{+1, -1, 0\}$ value set is adopted for the classification of an individual with respect to a query concept $Q$ and where the three values denote, respectively, membership, non-membership, and uncertainty. Hence, the task is cast as follows: given a query concept $Q$, determine the membership of an instance $x_q$ through the NN procedure (see Eq. 3) where $V = \{-1, 0, +1\}$ and the hypothesis function values for the training instances are determined as:

$$h_Q(x) = \begin{cases} +1 & \mathcal{K} \models Q(x) \\ -1 & \mathcal{K} \models \neg Q(x) \\ \;\;0 & \textit{otherwise.} \end{cases}$$

That is, the value of $h_Q$ for the training instances is determined by logical entailment (denoted $\models$) of the corresponding assertion from the knowledge base. Alternatively, a look-up in the ABox of the knowledge base could be considered, thus obtaining a classification process less complex but also possibly less accurate.

For measuring the similarity between individuals, a totally semantic and language independent family of dissimilarity measures has been used [5]. It is based on the idea of comparing the semantics of the input individuals along a number of dimensions represented by a committee of concept descriptions, say $\mathsf{F} = \{F_1, F_2, \ldots, F_m\}$, which stands as a group of discriminating *features* expressed in the OWL-DL sub-language taken into account. It is formally defined as follows [5]:

**Definition 5 (family of measures).** Let $KB = (\mathcal{T}, \mathcal{A})$ be a knowledge base. Given a set of concept descriptions $\mathsf{F} = \{F_1, F_2, \ldots, F_m\}$, corresponding weights $w_1, \ldots, w_m$, and $p > 0$, a family of dissimilarity functions $d_p^{\mathsf{F}} : \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mapsto [0, 1]$ is defined by:

$$\forall a, b \in \mathsf{Ind}(\mathcal{A})\colon \quad d_p^{\mathsf{F}}(a, b) := \frac{1}{|\mathsf{F}|} \left[ \sum_{i=1}^{|\mathsf{F}|} w_i \mid \delta_i(a, b) \mid^p \right]^{1/p} ,$$

where the dissimilarity function $\delta_i$ ($i \in \{1, \ldots, m\}$) is defined as follows:

$$\forall a, b \in \mathsf{Ind}(\mathcal{A})\colon \quad \delta_i(a, b) = \begin{cases} 0 & F_i(a) \in \mathcal{A} \wedge F_i(b) \in \mathcal{A} \\ 1 & F_i(a) \in \mathcal{A} \wedge \neg F_i(b) \in \mathcal{A} \text{ or} \\ & \neg F_i(a) \in \mathcal{A} \wedge F_i(b) \in \mathcal{A} \\ 1/2 & \mathit{otherwise}. \end{cases}$$

An alternative definition for the projections requires the entailment of an assertion (instance-checking) rather than the simple ABox look-up; this can make the measure more accurate yet more complex to compute. Moreover, using instance checking, induction is performed on top of deduction, thus making it a kind of completion of deductive reasoning.

As for the weights $w_i$ employed in the family of measures, they should reflect the impact of the single feature concept $F_i$ relative to the overall dissimilarity. This is determined by the quantity of information conveyed by a feature, which is measured as its entropy. Namely, the extension of a feature $F_i$ relative to the whole domain of objects may be probabilistically quantified as $P_{F_i} = |F_i^{\mathcal{I}}|/|\Delta^{\mathcal{I}}|$ (relative to the canonical interpretation $\mathcal{I}$). This can be roughly approximated by $|\mathsf{retrieval}(F_i)|/|\mathsf{Ind}(\mathcal{A})|$. Hence, considering also the probability $P_{\neg F_i}$ related to its negation and the one related to the unclassified individuals (relative to $F_i$), denoted $P_U$, we may give an entropic measure for the feature:

$$H(F_i) = -\left(P_{F_i} \log(P_{F_i}) + P_{\neg F_i} \log(P_{\neg F_i}) + P_U \log(P_U)\right) .$$

The measures strongly depend on $\mathsf{F}$. Here, we make the assumption that the feature-set $\mathsf{F}$ represents a sufficient number of (possibly redundant) features that are able to discriminate really different individuals. However, an optimal discriminating feature set could be learned [7]. Experimentally, we obtained good results by using the very set of both primitive and defined concepts found in the knowledge base [5].

**Measuring the Likelihood of an Answer.** The inductive inference made by the procedure shown above is not guaranteed to be deductively valid. Indeed, inductive inference naturally yields a certain degree of uncertainty. So, from a more general perspective, the main idea behind the above inductive inference for Semantic Web search is closely related to the idea of using probabilistic ontologies to increase the precision and the recall of querying databases and of information retrieval in general. But, rather than learning probabilistic ontologies from data, representing probabilistic ontologies, and reasoning with probabilistic ontologies, we directly use the data in the inductive inference step.

In order to measure the likelihood of the decision made by the inductive procedure (individual $x_q$ belongs to the query concept denoted by value $v$ maximizing the argmax argument in Eq. 3), given the $k$-nearest training individuals in $NN(x_q) = \{x_1, \ldots, x_k\}$, the quantity that determined the decision should be normalized by dividing it by the sum of such arguments over the (three) possible values:

$$l(\mathit{class}(x_q) = v | NN(x_q)) = \frac{\sum_{i=1}^{k} w_i \cdot \delta(v, h_Q(x_i))}{\sum_{v' \in V} \sum_{i=1}^{k} w_i \cdot \delta(v', h_Q(x_i))} . \tag{4}$$

Hence, the likelihood of the assertion $Q(x_q)$ corresponds to the case when $v = +1$. The computed likelihood can be used for building a probabilistic ABox (which is a collection of pairs, each consisting of a classical ABox axiom and a probability value).

## 6 Inconsistencies, Noise, and Incompleteness

In this section, we illustrate the main advantages of using inductive reasoning in Semantic Web search, namely, that inductive reasoning (differently from deductive reasoning) can handle inconsistencies, noise, and incompleteness in Semantic Web knowledge bases, which are all very likely to occur when knowledge bases are stored in a distributed and heterogeneous fashion, like on the Web.

**Inconsistencies.** Since our inductive method is based on the majority vote of the individuals in the neighborhood, it may be able to give a correct classification even in the case of inconsistent knowledge bases. This aspect is illustrated by the following example.

*Example 6.* Consider the description logic knowledge base $KB = (\mathcal{T}, \mathcal{A})$ that consists of the following TBox $\mathcal{T}$ and ABox $\mathcal{A}$:

$$\begin{aligned}
\mathcal{T} = \{&\mathit{Man} \equiv \mathit{Male} \sqcap \mathit{Human};\ \mathit{Professor} \equiv \mathit{Person} \sqcap \exists \mathit{abilitatedTo.Teaching} \sqcap \\
&\exists \mathit{isSupervisorOf.PhDThesis} \sqcap \mathit{Researcher};\ \mathit{Researcher} \equiv \mathit{GraduatePerson} \sqcap \\
&\exists \mathit{worksFor.ResearchInstitute} \sqcap \neg\exists \mathit{isSupervisorOf.PhDThesis}; \ldots\}\,; \\
\mathcal{A} = \{&\mathit{Professor}(\mathit{Franz});\ \mathit{isSupervisorOf}(\mathit{Franz}, \mathit{DLThesis});\ \mathit{Professor}(\mathit{John}); \\
&\mathit{isSupervisorOf}(\mathit{John}, \mathit{RoboticsThesis});\ \mathit{Professor}(\mathit{Flo});\ \mathit{isSupervisorOf}(\mathit{Flo}, \mathit{MLThesis}); \\
&\mathit{Researcher}(\mathit{Nick});\ \mathit{Researcher}(\mathit{Ann});\ \mathit{isSupervisorOf}(\mathit{Nick}, \mathit{SWThesis}); \ldots\}\,.
\end{aligned}$$

Actually, *Nick* is a *Professor*, indeed, he is the supervisor of a PhD thesis in $\mathcal{A}$. However, by human mistake, he is asserted to be a *Researcher* in $\mathcal{A}$, and by the axiom for *Researcher* in $\mathcal{T}$, he cannot be the supervisor of any PhD thesis. Hence, $KB$ is inconsistent, and thus a deductive reasoner cannot answer whether *Nick* is a *Professor* or not (since everything can be deduced from an inconsistent knowledge base). On the contrary, by inductive reasoning, it is highly probable that the returned classification result is that *Nick* is an instance of *Professor*. This is because the most similar individuals are *Franz*, *John*, and *Flo*, and all of them vote for the concept *Professor*.

**Noise.** Inductive reasoning may also be able to give a correct classification in the presence of noise in a knowledge base (containing, e.g., incorrect concept and/or role membership assertions), which is illustrated by the following example.

*Example 7.* Consider the description logic knowledge base $KB = (\mathcal{T}', \mathcal{A})$, where the ABox $\mathcal{A}$ is as in Example 6 and the TBox $\mathcal{T}'$ is obtained from the TBox $\mathcal{T}$ of Example 6 by replacing the axiom for *Researcher* by the following axiom:

$$\mathit{Researcher} \equiv \mathit{GraduatePerson} \sqcap \exists \mathit{worksFor.ResearchInstitute}\,.$$

Again, *Nick* is actually a *Professor*, but by human mistake asserted to be a *Researcher* in $KB$. But due to the slightly modified axiom for *Researcher*, there is no inconsistency in $KB$ anymore. By deductive reasoning, however, *Nick* turns out to be a *Researcher*, whereas by inductive reasoning, it is highly probable that the returned classification result is that *Nick* is an instance of *Professor*, as above, because the most similar individuals are *Franz*, *John*, and *Flo*, and all of them vote for the concept *Professor*.

**Incompleteness.** Clearly, inductive reasoning may also be able to give a correct classification in the presence of incompleteness in a knowledge base. That is, inductive reasoning is not necessarily deductively valid, and may produce new knowledge.

*Example 8.* Consider the description logic knowledge base $KB = (\mathcal{T}', \mathcal{A}')$, where the TBox $\mathcal{T}'$ is as in Example 7 and the ABox $\mathcal{A}'$ is obtained from the ABox $\mathcal{A}$ of Example 6 by removing the axiom $\mathit{Researcher}(\mathit{Nick})$. Then, the resulting knowledge base is neither inconsistent nor noisy, but it is now incomplete. Nonetheless, by the same line of argumentation as in Examples 6 and 7, it is highly probable that the classification result by inductive reasoning is that *Nick* is an instance of *Professor*.

## 7 Implementation and Experiments

In this section, we describe our prototype implementation for a semantic desktop search engine. Furthermore, we report on very positive experimental results on the precision and the recall under inductively vs. deductively completed semantic annotations.

**Implementation.** We have implemented a prototype for a semantic desktop search engine. We have realized both a deductive and an inductive version of the offline inference step for generating the completed semantic annotation for every considered resource. The deductive version uses PELLET[1], while the inductive one is based on the $k$-NN technique, integrated with an entropic measure, as proposed in Section 5. Specifically, each individual $i$ of a Semantic Web knowledge base is classified relative to all atomic concepts and all restrictions $\exists R^{-}.\{i\}$ with roles $R$. The parameter $k$ was set to $\log(|\mathsf{Ind}(\mathcal{A})|)$, where $\mathsf{Ind}(\mathcal{A})$ stands for all individuals in the knowledge base. The simpler distances $d_1^{\mathsf{F}}$ were employed, using all the atomic concepts in the knowledge base for determining the set $\mathsf{F}$.

**Precision and Recall of Inductive Semantic Web Search.** We next give an experimental comparison between Semantic Web search under inductive and under deductive reasoning. We do this by providing the precision and the recall of the latter vs. the former. Our experimental results with queries relative to the FINITE-STATE-MACHINE (FSM) and the SURFACE-WATER-MODEL (SWM) ontology from the Protégé Ontology Library[2] are summarized in Table 1. For example, Query (8) asks for all transitions having no target state, while Query (16) asks for all numerical models having either the domain "lake" and public availability, or the domain "coastalArea" and commercial availability. The experimental results in Table 1 essentially show that the answer sets under inductive reasoning are very close to the ones under deductive reasoning.

## 8 Summary and Outlook

We have presented a combination of Semantic Web search as presented in [8] with an inductive reasoning technique, based on similarity search [11] for retrieving the resources that likely belong to a query concept [5]. As a crucial advantage, the new approach to Semantic Web search allows for handling inconsistencies, noise, and incompleteness, which are very likely in distributed and heterogeneous environments, such as the Web. We have also reported on a prototype implementation and very positive experimental results on the precision and the recall of the new inductive approach to Semantic Web search.

[1] http://www.mindswap.org

[2] http://protegewiki.stanford.edu/index.php/Protege_Ontology_Library

**Table 1.** Precision and recall of inductive vs. deductive Semantic Web search.

| | Onto-<br>logy | **Query** | No. Results<br>Deduction | No. Results<br>Induction | No. Correct Results<br>Induction | **Precision<br>Induction** | **Recall<br>Induction** |
|---|---|---|---|---|---|---|---|
| 1 | FSM | $\mathit{State}(x)$ | 11 | 11 | 11 | 1 | 1 |
| 2 | FSM | $\mathit{StateMachineElement}(x)$ | 37 | 37 | 37 | 1 | 1 |
| 3 | FSM | $\mathit{Composite}(x) \wedge \mathit{hasStateMachineElement}(x, \mathit{accountDetails})$ | 1 | 1 | 1 | 1 | 1 |
| 4 | FSM | $\mathit{State}(y) \wedge \mathit{StateMachineElement}(x) \wedge \mathit{hasStateMachineElement}(x, y)$ | 3 | 3 | 3 | 1 | 1 |
| 5 | FSM | $\mathit{Action}(x) \vee \mathit{Guard}(x)$ | 12 | 12 | 12 | 1 | 1 |
| 6 | FSM | $\exists y, z\, (\mathit{State}(y) \wedge \mathit{State}(z) \wedge \mathit{Transition}(x) \wedge \mathit{source}(x, y) \wedge \mathit{target}(x, z))$ | 11 | 2 | 2 | 1 | 0.18 |
| 7 | FSM | $\mathit{StateMachineElement}(x) \wedge \mathit{not}\ \exists y\, (\mathit{StateMachineElement}(y) \wedge$<br>$\mathit{hasStateMachineElement}(x, y))$ | 34 | 34 | 34 | 1 | 1 |
| 8 | FSM | $\mathit{Transition}(x) \wedge \mathit{not}\ \exists y\, (\mathit{State}(y) \wedge target(x, y))$ | 0 | 5 | 0 | 0 | 1 |
| 9 | FSM | $\exists y\, (\mathit{StateMachineElement}(x) \wedge \mathit{not\ hasStateMachineElement}(x,$<br>$\mathit{accountDetails}) \wedge \mathit{hasStateMachineElement}(x, y) \wedge \mathit{State}(y))$ | 2 | 2 | 2 | 1 | 1 |
| 10 | SWM | $\mathit{Model}(x)$ | 56 | 56 | 56 | 1 | 1 |
| 11 | SWM | $\mathit{Mathematical}(x)$ | 64 | 64 | 64 | 1 | 1 |
| 12 | SWM | $\mathit{Model}(x) \wedge \mathit{hasDomain}(x, \mathit{lake}) \wedge \mathit{hasDomain}(x, \mathit{river})$ | 9 | 9 | 9 | 1 | 1 |
| 13 | SWM | $\mathit{Model}(x) \wedge \mathit{not}\ \exists y\, (\mathit{Availability}(y) \wedge \mathit{hasAvailability}(x, y))$ | 11 | 11 | 11 | 1 | 1 |
| 14 | SWM | $\mathit{Model}(x) \wedge \mathit{hasDomain}(x, \mathit{river}) \wedge \mathit{not\ hasAvailability}(x, \mathit{public})$ | 2 | 8 | 0 | 0 | 0 |
| 15 | SWM | $\exists y\, (\mathit{Model}(x) \wedge \mathit{hasDeveloper}(x, y) \wedge \mathit{University}(y))$ | 1 | 1 | 1 | 1 | 1 |
| 16 | SWM | $\mathit{Numerical}(x) \wedge \mathit{hasDomain}(x, \mathit{lake}) \wedge \mathit{hasAvailability}(x, \mathit{public}) \vee$<br>$\mathit{Numerical}(x) \wedge \mathit{hasDomain}(x, \mathit{coastalArea}) \wedge$<br>$\mathit{hasAvailability}(x, \mathit{commercial})$ | 12 | 9 | 9 | 1 | 0.75 |

In the future, we aim especially at extending the desktop implementation to a real Web implementation, using existing search engines, such as Google. Another interesting topic is to explore how search expressions that are formulated as plain natural language sentences can be translated into the ontological conjunctive queries of our approach. It would also be interesting to investigate the use of probabilistic ontologies rather than classical ones.

**Acknowledgments.** Georg Gottlob's work was supported by the EPSRC grant Number EP/E010865/1 "Schema Mappings and Automated Services for Data Integration." Georg Gottlob, whose work was partially carried out at the Oxford-Man Institute of Quantitative Finance, gratefully acknowledges support from the Royal Society as the holder of a Royal Society-Wolfson Research Merit Award. Thomas Lukasiewicz's work was supported by the German Research Foundation (DFG) under the Heisenberg Programme.

## References

[1] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.

[2] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Sci. Am.*, 284:34–43, 2001.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1–7):107–117, 1998.

[4] P.-A. Chirita, S. Costache, W. Nejdl, and S. Handschuh. P-TAG: Large scale automatic generation of personalized annotation TAGs for the Web. In *Proc. WWW-2007*.

[5] C. d'Amato, N. Fanizzi, and F. Esposito. Query answering and ontology population: An inductive approach. In *Proc. ESWC-2008*.

[6] L. Ding, T. W. Finin, A. Joshi, Y. Peng, R. Pan, and P. Reddivari. Search on the Semantic Web. *IEEE Computer*, 38(10):62–69, 2005.

[7] N. Fanizzi, C. d'Amato, and F. Esposito. Induction of classifiers through non-parametric methods for approximate classification and retrieval with ontologies. *International Journal of Semantic Computing*, 2(3):403–423, 2008.

[8] B. Fazzinga, G. Gianforme, G. Gottlob, and T. Lukasiewicz. From Web search to Semantic Web search. Technical Report INFSYS RR-1843-08-11, Institut für Informationssysteme, TU Wien, November 2008. http://www.kr.tuwien.ac.at/research/reports/rr0811.pdf.

[9] R. V. Guha, R. McCool, and E. Miller. Semantic search. In *Proc. WWW-2003*.

[10] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. Data Semantics*, 10:133–173, 2008.

[11] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search — The Metric Space Approach*, *Advances in Database Systems* 32. Springer, 2006.

# Evidential Nearest-Neighbors Classification for Inductive ABox Reasoning

Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito

Dipartimento di Informatica, Università degli studi di Bari
Campus Universitario, Via Orabona 4, 70125 Bari, Italy
{fanizzi,claudia.damato,esposito}@di.uniba.it

**Abstract.** In the line of our investigation on inductive methods for Semantic Web reasoning, we propose an alternative way for approximate ABox reasoning based on the analogical principle of the nearest-neighbors. Once neighbors of a test individual are selected, a combination rule descending from the Dempster-Shafer theory can join together the evidence provided by the neighbor individuals. We show how to exploit the procedure for determining unknown class- and role-memberships or fillers for datatype properties which may be the basis for many further ABox inductive reasoning algorithms.

## 1 Introduction

In the context of reasoning in the Semantic Web (SW), a growing interest is being committed to alternative procedures extending the standard methods so that they can deal with the various facets of uncertainty related with Web reasoning [1]. Extensions of the classic probability measures [2] offer alternative ways to deal with inherent uncertainty of the knowledge bases (KBs) in the SW. Particularly, belief and plausibility measures adopted in the *Dempster-Shafer Theory of Evidence* [3] have been exploited as means for dealing with incompleteness [4] and also inconsistency [5], which may arise from the aggregation of data and metadata on a large and distributed scale. In this work we undertake again the inductive point of view. Indeed, in many SW domains a very large number of assertions can potentially be true but often only a small number of them is known to be true or can be inferred to be true. So far the application of combination rules related to the Dempster-Shafer theory has concerned the induction of metrics which are essential for all similarity-based reasoning methods [4]. One of the applications of such measures was related to the prediction of assertions through nearest neighbor procedures. Recently a general-purpose evidential nearest neighbor procedure based on the Dempster-Shafer combination rule has been proposed [6]. In this work this method is extended to the specific case of semantic KBs through a more epistemically appropriate combination procedure [7]. In the perspective of inductive methods, the need for a definition of a semantic similarity measure for *individuals* arises, that is a problem that so far received less attention in the literature compared to the measures for concepts.

Recently proposed dissimilarity measures for individuals in specific languages founded in *Description Logics* [8] turned out to be practically effective for the targeted inductive tasks [9], however they are still based on structural criteria so that they can hardly scale to more complex languages. We devised families of dissimilarity measures for semantically annotated resources, which can overcome the aforementioned limitations [10, 11]. Our measures are mainly based on the Minkowski's norms for Euclidean spaces induced by means of a method developed in the context of relational *machine learning* [12]. Namely, the measures are based on the degree of discernibility of the input individuals with respect to a given context [13] (or committee of features), which are represented by concept descriptions expressed in the language of choice.

The main contributions of this work regard the extension of a framework for the classification of individuals through a prediction procedure based on evidence theory and similarity. In particular we propose using Yager's rule of combination and exploiting the mentioned families of metrics defined for individuals in ontologies. This allows for measuring the confirmation of the truth of candidate assertions. The prediction of the values (related to class-membership or datatype and object properties) may have plenty of applications in uncertainty reasoning with ontologies.

The remainder of the paper is organized as follows. In the next section (§2), distance measures that shall be utilized for selecting neighbor individuals are introduced. Then (§3), the basics of the Dempster-Shafer theory and a nearest-neighbor procedure based on an alternative rule of combination are recalled. Hence (§4) we present the applications of the method to the problems of determining the class- or role-membership of individuals w.r.t. given query concepts / roles as well as the prediction of fillers for datatype properties. Relevant related work are discussed in (§5) and we conclude (§6) proposing extensions and applications of these methods in further works.

## 2 Dissimilarity Measures for Individuals

Since the reasoning method to be presented in the following is intended to be general purpose, no specific language will be assumed in the following for resources, concepts (classes) and their properties. It suffices to consider a generic representation that can be mapped to some Description Logic language with the standard model-theoretic semantics (see [8] for a thorough reference).

A *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ comprises a *TBox* $\mathcal{T}$ and an *ABox* $\mathcal{A}$. $\mathcal{T}$ is a set of axioms concerning the (partial) definition of concepts (and roles) through class (role) expressions. $\mathcal{A}$ contains assertions (ground facts) concerning the world state. The set of the individuals occurring in $\mathcal{A}$ will be denoted with $\mathsf{Ind}(\mathcal{A})$. Each individual can be assumed to be identified by its own URI (it is useful in this context to make the *unique names assumption*).

Similarity-based tasks, such as individual classification, retrieval, and clustering require language-independent measures for individuals whose definition can capture semantic aspects of their occurrence in the knowledge base [10, 11].

For our purposes, we need functions to assess the similarity of individuals. However individuals do not have an explicit syntactic (or algebraic) structure that can be compared (unless one resorts to language-specific notions [9], such as the *most specific concept* [8]). Focusing on the semantic level, the leading idea may be that, similar individuals should behave similarly w.r.t. the same concepts. A way for assessing the similarity of individuals in a knowledge base can be based on the comparison of their semantics along a number of dimensions represented by a set of concept descriptions (henceforth referred to as the *committee* or *context* [13]). Specifically, the measure may compare individuals on the grounds of their behavior w.r.t. a given context, say $\mathsf{C} = \{C_1, C_2, \ldots, C_m\}$, which stands as a group of discriminating relevant concepts (*features*) expressed in the considered language. We begin with defining the behavior of an individual w.r.t. a certain concept in terms of projecting it in this dimension: Given a concept $C_i \in \mathsf{C}$, the related *projection function* $\pi_i : \mathsf{Ind}(\mathcal{A}) \mapsto \{0, \frac{1}{2}, 1\}$ is defined:

$$\forall a \in \mathsf{Ind}(\mathcal{A}) \qquad \pi_i(a) = \begin{cases} 1 & \mathcal{K} \models C_i(a) \\ 0 & \mathcal{K} \models \neg C_i(a) \\ \frac{1}{2} & \text{otherwise} \end{cases}$$

The case of $\pi_i(a) = \frac{1}{2}$ corresponds to the case when a reasoner cannot give the truth value for a certain membership query. This is due to the *Open World Assumption* normally made in Semantic Web reasoning. Hence, as in the classic probabilistic models, uncertainty may be coped with by considering a uniform distribution over the possible cases. Further ways to approximate these values in case of uncertainty are investigated in [4].

The discernibility functions related to the context w.r.t. which two input individuals are compared are defined as follows. Given a feature concept $C_i \in \mathsf{C}$, the related *discernibility function* $\delta_i : \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mapsto [0,1]$ is defined as:

$$\forall (a,b) \in \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \qquad \delta_i(a,b) = |\pi_i(a) - \pi_i(b)|$$

The discernibility function $\delta_i$ assigns 0 if the two individuals $a$ and $b$ have the same behavior w.r.t. $C_i$, that is if they are both instance of $C_i$ or both instance of $\neg C_i$ or nothing is known about this. This is because, if $a$ and $b$ have the same bahavior w.r.t. $C_i$ then there are no other information for discriminating them.

Finally, a family of dissimilarity measures for individuals that is inspired to the Minkowski's metrics can be defined [10, 11]: Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a knowledge base. Given a context $\mathsf{C}$ and a related vector of weights $\boldsymbol{w}$, a family of dissimilarity measures $\{d_p^{\mathsf{C}}\}_{p \in \mathbb{N}}$, $d_p^{\mathsf{C}} : \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mapsto [0,1]$ is defined as follows:

$$\forall (a,b) \in \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \qquad d_p^{\mathsf{C}}(a,b) = \left[ \sum_{C_i \in \mathsf{C}} w_i \delta_i(a,b)^p \right]^{\frac{1}{p}}$$

The effect of the weights[1] is to normalize w.r.t. the other features involved. Obviously these measures are not absolute, then they should be also considered

[1] A possible way for determining the $w_i$ is to assign a high value if the corresponding feature concept reflects high *information content*, low value otherwise (see [10] for more details).

w.r.t. the context of choice, hence comparisons across different contexts may not be meaningful. Larger contexts are likely to decrease the measures because of the normalizing factor yet these values is affected also by the degree of redundancy of the features employed. In other works the choice of the weights is done according to variance or entropy associated to the various concepts in the context [10, 11].

Compared to other proposed measures [14, 9, 15], the presented functions do not depend on the constructors of a specific language, rather they require only (retrieval or) instance-checking for computing the projections through class-membership queries to the knowledge base. The complexity of measuring the dissimilarity of two individuals depends on the complexity of such inferences (see [8], Ch. 3). Note also that the projections that determine the measure can be computed (or derived from statistics maintained on the knowledge base) before the actual distance application, thus determining a speed-up in the computation of the measure. This is very important for algorithms that massively use this distance, such as instance-based methods.

One should assume that $\mathsf{C}$ represents a set of (possibly redundant) features that are able to discriminate individuals that are actually different. The choice of the concepts to be included (a *feature selection* problem [12]) may be crucial. Therefore, specific optimization algorithms founded in *randomized search* have been devised which are able to find optimal choices of discriminating contexts [10, 11]. However, the results obtained so far with knowledge bases drawn from ontology libraries showed that (a selection) of the primitive and defined concepts are often sufficient to induce sufficiently discriminating measures.

## 3 Evidence-Theoretic Nearest-Neighbor Prediction

In this section the basics of the theory of evidence and combination rules [3] are recalled then a nearest neighbor classification procedure based on the rule of combination [6] is extended in order to perform prediction of unobserved values (related to datatype properties or also class-membership).

### 3.1 Basics of the Evidence Theory

In the Dempster-Shafer theory, a *frame of discernment* $\Omega$ is defined as the set of all hypotheses in a certain domain. Particularly, in a classification problem it is the set of all possible classes. A *basic belief assignment* (BBA) is a function $m$ that defines a mapping $m : 2^{\Omega} \mapsto [0, 1]$ verifying: $\sum_{A \in \Omega} m(A) = 1$. Given a certain piece of evidence, the value of the BBA for a given set $A$ expresses a measure of belief that is committed exactly to $A$. The quantity $m(A)$ pertains only to $A$ and does not imply any additional claims about any of its subsets. If $m(A) > 0$, then $A$ is called a *focal element* for $m$.

The BBA $m$ cannot be considered a proper probability measure: it is defined over $2^{\Omega}$ instead of $\Omega$ and it does not require the properties of monotone measures [2]. The BBA $m$ and its associated focal elements define a *body of evidence*, from which a *belief function Bel* and a *plausibility function Pl* can be derived as mappings from $2^{\Omega}$ to $[0, 1]$. For a given $A \subseteq \Omega$, the *belief* in $A$, denoted $Bel(A)$, represents a measure of the total belief committed to $A$ given the available evidence. $Bel$ is defined as follows:

$$\forall A \in 2^{\Omega} \qquad Bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B) \tag{1}$$

Analogously, the plausibility of $A$, denoted $Pl(A)$, represents the amount of belief that could be placed in $A$, if further information became available. $Pl$ is defined as follows:

$$\forall A \in 2^{\Omega} \qquad Pl(A) = \sum_{B \cap A \neq \emptyset} m(B) \tag{2}$$

It is easy to see that: $Pl(A) = Bel(\Omega) - Bel(\bar{A})$. Moreover $m(\emptyset) = 1 - Bel(\Omega)$ and for each $A \neq \emptyset$: $m(A) = \sum_{B \subseteq A} (-1)^{|A \setminus B|} Bel(B)$. Using these equations, knowing just one function among $m$, $Bel$, and $Pl$ allows to derive the others.

The Dempster-Shafer rule of combination [3] is an operation for pooling evidence from a variety of sources. This rule aggregates independent bodies of evidence defined within the same frame of discernment into one body of evidence. Let $m_1$ and $m_2$ be two BBAs. The new BBA obtained by combining $m_1$ and $m_2$ using the rule of combination, $m_{12}$ is the orthogonal sum of $m_1$ and $m_2$. Generally, the normalized version of the rule is used:

$$\forall A \in 2^{\Omega} \setminus \{\emptyset\} \qquad m_{12}(A) = (m_1 \oplus m_2)(A) = \frac{\sum_{B \cap C = A} m_1(B)\, m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B)\, m_2(C)}$$

(and $m_{12}(\emptyset) = 0$) where the numerator $(1 - c)$ normalizes the values of the combined BBA w.r.t. the amount of conflict $c$ between $m_1$ and $m_2$.

Different evidence fusion rules have been proposed [2]. A more epistemologically sound combination rule [7] for our purposes places the probability mass related to the conflict between the BBAs to the case of maximal ignorance.

$$\forall A \in 2^{\Omega} \qquad m_{12}(A) = \begin{cases} \sum_{B \cap C = A} m_1(B)\, m_2(C) & A \neq \Omega \wedge A \neq \emptyset \\ m_1(\Omega)\, m_2(\Omega) + c & A = \Omega \\ 0 & A = \emptyset \end{cases}$$

This means that the conflict between the two sources of evidence is not hidden, but it is explicitly recognized as a contributor to ignorance.

Due to the associativity and commutativity of the operations involved, it is easy to prove that the resulting combination operator is associative and commutative, and admits the vacuous BBA ($\Omega$ unique focal set) as neutral element.

### 3.2 The Nearest Neighbors Procedure

Let us consider the finite set of instances $X$ and a finite set of integers $V \subseteq \mathbb{Z}$ to be used as labels (which may correspond to disjoint classes or distinct attribute values). The available information is assumed to consist in a training set $\mathsf{TrSet} =$

$\{(x_1, v_1), \ldots, (x_M, v_M)\} \subseteq \mathsf{Ind} \times V$ of single-labeled instances (*examples*). In our case, $X = \mathsf{Ind}(\mathcal{A})$, the set of individual names occurring in the ontology.

Let $x_q$ be a new individual to be classified on the basis of its nearest neighbors in $\mathsf{TrSet}$. Let $N_k(x_q) = \{(x_{o(j)}, v_{o(j)}) \mid j = 1, \ldots, k\}$ be the set of the $k$ nearest neighbors of $x_q$ in $\mathsf{TrSet}$ sorted by a function $o(\cdot)$ depending on an appropriate metric $d$ which can be applied to ontology individuals (e.g. one of the measures in the family defined in the previous section §2).

Each pair $(x_i, v_i) \in N_k(x_q)$ constitutes a distinct item of evidence regarding the value to be predicted for $x_q$. If $x_q$ is close to $x_i$ according to $d$, then one will be inclined to believe that both instances are associated to the same value, while when $d(x_q, x_i)$ increases, this belief decreases and that leads to a situation of almost complete ignorance concerning the value to be predicted for $x_q$.

Consequently, each $(x_i, v_i) \in N_k(x_q)$ may induce a BBA $m_i$ over $V$ which can be defined as follows [6]:

$$\forall A \in 2^V \qquad m_i(A) = \begin{cases} \lambda\sigma(d(x_q, x_i)) & A = \{v_i\} \\ 1 - \lambda\sigma(d(x_q, x_i)) & A = V \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $\lambda \in ]0, 1[$ is a parameter and $\sigma(\cdot)$ is a decreasing function such that $\sigma(0) = 1$ and $\lim_{d \to \infty} \sigma(d) = 0$ (e.g. $\sigma(d) = \exp(-\gamma d^n)$ with $\gamma > 0$ and $n \in \mathbb{N}$). The values of the parameters can be determined heuristically.

Considering each training individual in $N_k(x_q)$ as an separate source of evidence, $k$ BBAs $m_j$ are obtained. These can be pooled by means of the rule of combination leading to the aggregated BBA $m$ that synthesizes the final belief:

$$\bar{m} = \bigoplus_{j=1}^{k} m_j = m_1 \oplus \cdots \oplus m_k \tag{4}$$

In order to predict a value, functions $\overline{Bel}$ and $\overline{Pl}$ can be derived from $\bar{m}$ using the equations seen above, and the query individual $x_q$ is assigned the value in $V$ that maximizes the belief or plausibility:

$$v_q = \operatorname*{argmax}_{(x_i, v_i) \in N_k(x_q)} \overline{Bel}(\{v_i\}) \quad \text{or} \quad v_q = \operatorname*{argmax}_{(x_i, v_i) \in N_k(x_q)} \overline{Pl}(\{v_i\})$$

The former choice (select the hypothesis with the greatest degree of belief the most credible) corresponds to a *skeptical* viewpoint while the latter (select the hypothesis with the lowest degree of doubt the most plausible) is more *credulous.* The degree belief (or plausibility) of the predicted value provides also a way to compare the answers of an algorithm built on top of such analogical procedure. This is useful for tasks such as ranking, matchmaking, etc..

Finally, it is possible to combine the two measures *Bel* and *Pl* analogously to necessity (*Nec*) and possibility (*Pos*) in *Possibility Theory* (which can be considered a special case[2] of Dempster-Shafer theory). One can define a single

[2] Precisely, the body of evidence must contain *consonant* focal sets, i.e. when the set of focal elements is a nested family [2].

```
$ENN_k(x_q, \mathsf{TrSet}, V)$

1. Compute the neighbor set $N_k(x_q) \subseteq \mathsf{TrSet}$.
2. for each $i \leftarrow 1$ to $k$ do
       Compute $m_i$ (Eq. 3)
3. for each $v \in V$ do
       Compute $\overline{m}$ (Eq. 4) and derive $\overline{Bel}$ and $\overline{Pl}$ (Eqs. 1–2)
       Compute the confirmation $\overline{C}$ (Eq. 5) from $\overline{Bel}$ and $\overline{Pl}$
4. Select $v \in V$ that maximizes $\overline{C}$ (Eq. 6).
```

**Fig. 1.** The evidence nearest neighbor procedure.

measure of *confirmation* $C$, ranging in $[-1, +1]$, by means of a simple one-to-one transformation [2]:

$$\forall A \subseteq \Omega \qquad C(A) = Bel(A) + Pl(A) - 1 \tag{5}$$

Hence, denoted with $\overline{C}$ the combination of $\overline{Bel}$ and $\overline{Pl}$, the resulting rule for predicting the uncertain value for the test individual can be written as follows:

$$v_q = \operatorname*{argmax}_{(x_i, v_i) \in N_k(x_q)} \overline{C}(\{v_i\}) \tag{6}$$

Summing up, the procedure is as reported in Fig. 1:

It is worthwhile to note that the complexity of the method is polynomial in the number of instances in the $\mathsf{TrSet}$. If this set is compact and contains very prototypical individuals with plenty of related assertions, then the resulting predictions are likely to be accurate. Another source of complexity in the computations may be the number of values in $V$ which may yield a large number of subsets $2^{|V|}$ for which BBAs are to be computed. However this depends also on the kind of problem that is to be solved (e.g. in class membership detection $|V| = 2$). Moreover what really matters in the number of focal sets for each BBA which may be much less than $2^{|V|}$.

## 4 Assertion Prediction

The utility of the presented procedure when applied to ontology reasoning can be manifold. In the following we propose its employment in the inductive prediction of unknown values related to class-membership and datatype / object property fillers. This feature may be easily embedded in an ontology management system in order to help the knowledge engineers elicit assertions which may be not be derived from the knowledge base yet they can be rather made in analogy with the others [9].

In the following, the symbol $|\!\!\approx$ in expressions like $\mathcal{K} \;|\!\!\approx\; \alpha$ will denote the derivation of the assertion $\alpha$ from the knowledge base $\mathcal{K}$ obtained through an alternative procedure (like the evidence nearest neighbor presented in the previous section).

### 4.1 Class-Membership

Let us suppose a (query) concept $Q$ is given. In this case one may consider only examples made up of individuals with a definite class-membership leading to a binary problem with a set of values $V_Q = \{+1, -1\}$ denoting, resp., membership and non-membership w.r.t. the query concept. Alternatively, one may admit ternary problems with some labels set to 0 to explicitly denote an indefinite (uncertain) class-membership [9, 10]. We shall also consider the related training set $\mathsf{TrSet}_Q \subseteq \mathsf{Ind}(\mathcal{A}) \times V_Q$. The values of the labels $v_i$ for the training examples can be obtained through deductive reasoning (instance-checking) or specific facilities made available by the knowledge management systems [16].

Now to predict the class-membership value $v_q$ for some individual $x_q$ w.r.t. $Q$, it suffices to call the procedure $\mathit{ENN}_k(x_q, \mathsf{TrSet}_Q, V_Q)$ and decide on the grounds of the returned value. Thus in a binary setting ($V_Q = \{+1, -1\}$), one will either conclude that $\mathcal{K} \models\!\!\!\approx Q(x_q)$ or $\mathcal{K} \models\!\!\!\approx \neg Q(x_q)$ depending on the value that maximizes $\overline{C}$ in Eq. 6 (resp., $v_q = +1$ or $v_q = -1$). Moreover the value of the confirmation function which determined the returned value $v_q$ can be exploited for ranking the hits by comparing the strength of the inductive conclusions.

Adopting a ternary setting, it may turn out that the most likely value is $v_q = 0$ resulting in an uncertain case. One may force the choice among the values of $\overline{C}$ for $v_q = -1$ and $v_q = +1$, e.g. when the confirmation degree exceeds a some threshold.

The inductive procedure described above can be trivially exploited for performing the retrieval of a certain concept inductively. Given a certain concept $Q$, it would suffice to find all individuals $a \in \mathsf{Ind}(\mathcal{A})$ that are such that $\mathcal{K} \models\!\!\!\approx Q(a)$. The hits could be returned ranked by the respective confirmation value $\overline{C}(+1)$.

### 4.2 Datatype Fillers

In this case, let us suppose a certain (functional) datatype property $P$ is given and the problem is to predict its value for a certain test individual $a$ (which has to be supposed to be in its domain). The set of values $V_P$ may correspond to the (discrete and finite) range of the property or to its restriction to the observed values for the training instances: $V_P = \{v \in \mathit{range}(P) \mid \exists P(a, v) \in \mathcal{A}\}$. Different settings may be devised allowing for some special value(s) denoting the case of a yet unobserved value(s) for that property.

The related training set will be some $\mathsf{TrSet}_P \subseteq \mathit{domain}(P) \times V_P$, where $\mathit{domain}(P) \subseteq \mathsf{Ind}(\mathcal{A})$ is the set of individual names that have a known $P$-value in the knowledge base. Differently from the previous problem, datatype properties generally do not have a specific intensional definition in the knowledge base (except for the specification of domain and range), hence a mere look-up in the ABox should suffice to determine the $\mathsf{TrSet}$.

Now to predict the value in $V_P$ of the datatype property $P$ for some individual $a$, the method requires calling the procedure with $\mathit{ENN}_k(a, \mathsf{TrSet}_P, V_P)$. Thus in this setting, if $v_q$ is the value that maximizes Eq. 6 then we can write $\mathcal{K} \models\!\!\!\approx P(a, v_q)$. Also in this case the value of the confirmation function which determined choice of the value $v_q$ can be exploited for comparing the strength of an inductive conclusion to others.

In case of special settings with dummy values indicating unobserved values, when these are found to be the most credible among the others, a knowledge engineer should be contacted for the necessary changes to the ontology.

The inductive procedure described above can be trivially exploited for performing alternate forms of retrieval, e.g. finding all individuals with a certain value for the given property. Given a certain value $v$, it would suffice to find all individuals $a \in \mathsf{Ind}(\mathcal{A})$ that are such that $\mathcal{K} \approx\!\!\models P(a,v)$. Again, the hits could be returned ranked according to the respective confirmation value $\overline{C}(+1)$.

The limitation of treating only functional datatype properties may be overcome by considering a different way to assign the probability mass to BBAs than Eq. 3, including subsets of all possible values. Examples are to be constructed accordingly (labels will be chosen in $2^{V_P}$). Alternatively, more complex frames of discernment, e.g. $\Omega' = 2^{\Omega}$, so consider sets of values as possible fillers of the property. In all such settings the computation of the BBAs and descending measures would become of course much more complex and expensive, yet clever solutions (or approximations) proposed in the literature [6] may contribute to mitigate this problem.

### 4.3 Relationships among Individuals

In principle, a very similar setting may be used in order to establish the possibility that a certain test individual is related through some object property with some other individual [17, 18].

Since the set $\mathsf{Ind}(\mathcal{A})$ is finite (the target is not discovering relations with unseen individuals), one may want to find all individuals that are related to a test one through some object property, say $R$. The problem can be decomposed into smaller ones aiming at verifying whether $\mathcal{K} \approx\!\!\models R(a,b)$ holds:

```
for each $b \in \mathsf{Ind}(\mathcal{A})$ do
  for each $a \in \mathsf{Ind}(\mathcal{A})$ do
    $\mathsf{TrSet} \leftarrow \{(x,v) \mid x \in \mathsf{Ind}(\mathcal{A}) \setminus \{a\}, \mathit{if}\ \mathcal{K} \models R(x,b)\ \mathit{then}\ v \leftarrow +1\ \mathit{else}\ v \leftarrow -1\}$
    $v_b^R \leftarrow \mathit{ENN}_k(a, \mathsf{TrSet}, \{+1,-1\})$
    if $v_b^R = +1$ then
      return $\mathcal{K} \approx\!\!\models R(a,b)$
    else
      return $\mathcal{K} \approx\!\!\models \neg R(a,b)$
```

Note that, in the construction of the training sets, the inference $\mathcal{K} \models R(x,b)$ may turn out to be merely an ABox lookup operation for the given assertions (when roles are not intensionally defined in a proper RBox). Conversely, if an RBox is available (sometimes as a subset of the TBox) the values of the label for the training examples can be obtained through deductive reasoning (instance-checking) or the mentioned facilities made available by advanced reasoners or knowledge management systems [16].

This simple setting makes a sort of *closed-world assumption* in the decision of the induced assertions descending from the adoption of the binary value set and the composition of the TrSet. A more cautious setting would involve a ternary value set $V_R = \{-1, 0, +1\}$ which allows for an explicit treatment of those individuals $a$ for which $R(a, b)$ is not derivable (or just absent from the ABox). The final decision on the induced conclusion has to consider also this new possibility (e.g. using a threshold of confirmation for accepting likely assertions).

## 5 Related Work

The proposed method is related to those approaches devised to offer alternative ways of reasoning with ABoxes for eliciting hidden knowledge (regularities) in order to complete and populate the ontology with likely assertions even in the occurrence of incorrect parts, supposing this kind of *noise* is not systematic.

The tasks of ontology completion and population have often been tackled through formal methods (such as *formal concept analysis* [19]). Discovering new assertions (and related probabilities in a classical setting) is another related task for eliciting hidden knowledge in the ontologies. In [18] a machine learning method is proposed to estimate the truth of statements by exploiting regularities in the data. In [17] another statistical learning method for OWL-DL ontologies is proposed, combining a latent relational graphical model with Description Logic inference in a modular fashion. The probability of unknown role-assertions can be inductively inferred and known concept-assertions can be analyzed by clustering individuals.

Similarity-based reasoning with ontologies is the primary aim of this work which follows a number of related methods founded on dissimilarity measures for individuals in knowledge bases expressed in Description Logics [9, 10]. Mostly, they adopt some alternate form of the classic Nearest-Neighbor lazy learning scheme [12] in order to draw inductive conclusions that often cannot be deductively entailed by the knowledge bases.

Similar approaches based on lazy learning have been proposed that adopt generalized probability theories such as the Dempster-Shafer. In [6], which was a source of inspiration for this paper, the standard rule of combination is exploited in an evidence-theoretic classification procedure where labels were not assumed to be mutually exclusive. Rules of combination had been used in [4] in order to learn precise metrics to be exploited in a lazy learning setting like those mentioned above.

One of the most appreciated advantages of performing inductive ABox reasoning through these methods is that they can naturally handle inconsistent (and inherently incomplete) knowledge bases, especially when inconsistency is not systematic. In [5] a method for dealing with inconsistent ABoxes populated through information extraction is proposed: it constructs ad hoc belief networks for the conflicting parts in an ontology and adopts the Dempster-Shafer theory for assessing the confidence of the resulting assertions.

## 6 Concluding Remarks and Outlook

In the line of our investigation of inductive methods for Semantic Web reasoning, we have proposed an alternative way for approximate ABox reasoning based on the nearest-neighbors analogical principle. Once neighbors of a test individual are selected through some distance measures, a combination rule descending from the Dempster-Shafer theory can fuse the evidence provided by the various neighbor individuals. We have shown how to exploit the procedure for assertion prediction problems such as determining unknown class- or role-memberships as well as attribute-values which may be the basis for many ABox inductive reasoning algorithms. The method is being implemented so to allow an extensive experimentation on real ontologies.

Special settings to accommodate cases of uncertain or unobserved values are to be investigated. One promising extension of the method concerns the possibility of considering infinite sets of values $V$ following the studies [20, 2]. This would allow dealing with domains where the total amount of values is unknown (also due to the inherent nature of the Semantic Web). Moreover the predicted values often need not to be exclusive. Hence the prediction procedure would require an extension towards the consideration of sets of values instead of singletons.

As necessity and possibility measures are related to the belief measures (see note 2 at page 32) a natural extension may be towards the possibilistic theory and its calculus which is, in general, different from the Dempster-Shafer theory and calculus. Further possible extensions concern all other monotone measures such as the Sugeno $\lambda$-measures [2]. The extension towards the Possibility Theory is interesting also because of its parallelism with modal logics [20] and possibilistic extensions of Description Logics [21].

## References

1. Laskey, K., Laskey, K., Costa, P., Kokar, M., Martin, T., Lukasiewicz, T.: Uncertainty Reasoning for the World Wide Web. W3C Incubator Group. (2008) http://www.w3.org/2005/Incubator/urw3/XGR-urw3-20080331/.
2. Klir, G.: Uncertainty and Information. Wiley (2006)
3. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press (1976)
4. Fanizzi, N., d'Amato, C., Esposito, F.: Approximate measures of semantic dissimilarity under uncertainty. In da Costa, P., et al., eds.: Uncertainty Reasoning for the Semantic Web I. Volume 5327 of LNAI. Springer (2008) 355–372
5. Nikolov, A., Urea, V., Motta, E., de Roeck, A.: Using the Dempster-Shafer theory of evidence to resolve ABox inconsistencies. In da Costa, P., et al., eds.: Uncertainty Reasoning for the Semantic Web I. Volume 5327 of LNAI. Springer (2008) 143–160
6. Denoeux, T.: A k-nearest neighbor classication rule based on Dempster-Shafer theory. IEEE Transactions on Systems, Man and Cybernetics **25** (1995) 804–813
7. Yager, R.: On the Dempster-Shafer framework and new combination rules. Information Sciences **41** (1987) 93–137
8. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook. Cambridge University Press (2003)

9. d'Amato, C., Fanizzi, N., Esposito, F.: Analogical reasoning in description logics. In da Costa, P., et al., eds.: Uncertainty Reasoning for the Semantic Web I. Volume 5327 of LNAI. Springer (2008) 336–354
10. d'Amato, C., Fanizzi, N., Esposito, F.: Query answering and ontology population: An inductive approach. In Bechhofer, S., et al., eds.: Proceedings of the 5th European Semantic Web Conference, ESWC2008. Volume 5021 of LNCS., Springer (2008) 288–302
11. Fanizzi, N., d'Amato, C., Esposito, F.: Metric-based stochastic conceptual clustering for ontologies. Information Systems **34** (2009) 725–739
12. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning – Data Mining, Inference, and Prediction. Springer (2001)
13. Goldstone, R., Medin, D., Halberstadt, J.: Similarity in context. Memory and Cognition **25** (1997) 237–255
14. Borgida, A., Walsh, T., Hirsh, H.: Towards measuring similarity in description logics. In Horrocks, I., Sattler, U., Wolter, F., eds.: Working Notes of the International Description Logics Workshop. Volume 147 of CEUR Workshop Proceedings., Edinburgh, UK (2005)
15. d'Amato, C., Staab, S., Fanizzi, N.: On the influence of description logics ontologies on conceptual similarity. In Gangemi, A., Euzenat, J., eds.: Proceedings of the 16th Knowledge Engineering Conference, EKAW2008. Volume 5268 of LNAI., Springer (2008) 48–63
16. Horrocks, I., Li, L., Turi, D., Bechhofer, S.: The Instance Store: DL reasoning with large numbers of individuals. In Haarslev, V., Möller, R., eds.: Proceedings of the 2004 Description Logic Workshop, DL 2004. Volume 104 of CEUR Workshop Proceedings., CEUR (2004) 31–40
17. Rettinger, A., Nickles, M.: Infinite hidden semantic models for learning with OWL DL. In d'Amato, C., et al., eds.: Proceedings of 1st ESWC Workshop on Inductive Reasoning and Machine Learning for the Semantic Web, IRMLeS09. Volume 474 of CEUR Workshop Proceedings., Heraklion, Greece (2009)
18. Tresp, V., Huang, Y., Bundschus, M., Rettinger, A.: Materializing and querying learned knowledge. In d'Amato, C., et al., eds.: Proceedings of 1st ESWC Workshop on Inductive Reasoning and Machine Learning for the Semantic Web, IRMLeS09. Volume 474 of CEUR Workshop Proceedings., Heraklion, Greece (2009)
19. Baader, F., Ganter, B., Sertkaya, B., Sattler, U.: Completing description logic knowledge bases using formal concept analysis. In Veloso, M., ed.: Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India (2007) 230–235
20. Harmanec, D., Klir, G., Wang, Z.: Modal logic interpretation of Dempster-Shafer theory: An infinite case. International Journal of Approximate Reasoning **14** (1996) 81–93
21. Qi, G., Pan, J., Ji, Q.: A possibilistic extension of description logics. In Calvanese, D., et al., eds.: Working Notes of the 20th International Description Logics Workshop, DL2007. Volume 250 of CEUR Workshop Proceedings., Bressanone, Italy (2007) 435–442

# Ontology Granulation Through Inductive Decision Trees[1]

Bart Gajderowicz, Alireza Sadeghian

Ryerson University, Computer Science Department, 250 Victoria Street, Toronto, Ontario, Canada
{bgajdero@ryerson.ca, asadeghi@ryerson.ca}

The popularity of ontologies for representing the semantics behind many real-world domains has created a growing pool of ontologies on various topics. While different ontologists, experts, and organizations create the vast majority of ontologies, often for closed world systems, their domains frequently overlap in an open world system, such as the Semantic Web. These overlapping ontologies sometimes model similar or matching theories, that may be inconsistent. To assist in the reuse of these ontologies, this paper describes a technique for enriching manually created ontologies by supplementing them with inductively derived rules, and reducing the number of inconsistencies. The derived rules are translated from decision trees created by executing a tree based data mining algorithm with probability measures over the data being modeled. These rules can be used to revise the ontology adding a higher level of granularity, in order to identify possible similarities missed by the original ontologists. We then discuss how this may be applied to ontology matching. We demonstrate the application of our technique by presenting an example, and discuss how various data types may be treated to generalize the semantics of an ontology for an open world system.

**Keywords:** probabilistic ontology, ontology granulation, ontology matching, decision trees.

## 1. Introduction

In today's open community, more organizations are willing to share their data, in the hopes of improving their processes through collaboration. A problem arises when their internal, closed world, information and assumptions are un-interpretable in the open-world environment. Upper ontologies such as DOLCE [14], OpenCYC [22], and SUMO[23], have been used to serve as a place for defining general concepts, heavily based on natural language and common sense. Cross-references through such general concepts has been envisioned as helping in matching one ontology to another, promoting their reusability, assisting in automated inference and natural language processing [11]. Manual ontology creation and matching has been conducted by ontologists and subject matter experts, based on their experiences and context [12], but is time consuming and error prone [12].

We propose an algorithm for enhancing an existing ontology[2] with decision trees (DT) obtained from domain specific data, and refining observations made, for the purpose of increasing the probability of finding a match between ontologies. In previous work, ontologies have been utilized to build decision trees. As demonstrated in the development of the Ontology-driven Decision Tree (ODT) algorithm [29], ontologies provide ISA relations to link instances in the data with super-classes in the ontology. ODT considers an attribute's information gain, but modifies the decision tree by inserting the super-class of each instance from the ontology as a sub-node, instead of the actual instances. A similar approach to ODT was used in combination with user ratings to develop a recommender system called SemTree [5]. The advantage in using an ontology is that the key factor of the building process, the information-gain used to associate an attribute to a concept, is based on the attribute's semantic relation to that concept, in addition to its value as in traditional DTs. This paper proposes using those semantic relationships to create identification rules, in the form of DTs, to differentiate concepts from each other, based on their relationships in the ontology.

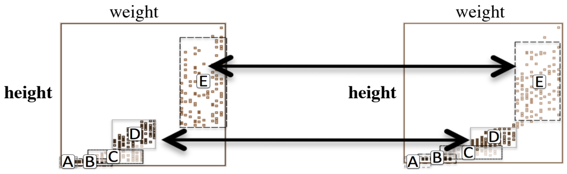
A possible domain where this is applicable is in scientific research, where the results are only as accurate as their underlying data. When qualifying collected specimens or observed phenomena, the researcher often relies on a combination of data-driven and theory-driven information [4]. In fields such as geology, qualifying various types of rock depends greatly on the specimens found and the geologist's knowledge

[1] This paper is a progress report about the 1st author's master's thesis.

[2] This paper targets ontologies which are represented by a direct acyclic graph (DAG) and compatible languages.

about the region, rock types, and properties which are infrequently observed but theoretically important. Due to personal bias, some theoretical knowledge may be used incorrectly due to incorrect qualification of the location, for example as a *lake* instead of *stream*. Brodaric et al. [4] observed that more consistent, and presumed correct, qualifications were exhibited using data-driven information, versus theory-based.

For example, the classification of *cat*, *tiger*, and *panther* as subclasses of *felinae* do not have enough non-lexical information to differentiate them from each other. The addition of physical attributes such as weight ranges or geographical locations may provide information which allows for differentiation. Further, attribute level information may be consistent amongst the instances observed by other ontologists, even when it does not apply to their domain. If so, it may be used to match these concepts[3] at a more detailed level based on a *learned* model from instance data [11], in the form of DTs, which are association with edges in the ontologies. As will be expanded on in Section 4, the consistency demonstrated between clusters in Figure 1 may be used to match the *classified* concepts from one ontology to another. In section 2 we give relevant background information on the covered topics, and describe how it may be used for ontology matching[4]. Section 3 gives a detailed definition of our contribution, the granulation algorithm. In Section 4 we expanded on the applicability of the algorithm, and summarize our findings in section 5.



**(a)** **(b)**

**Fig 1.** Classifying instances using concepts of different ontologies based on a pair of attributes *weight* and *height*, reveal similarity correlation between the same pair of attributes, in separate ontologies (a) and (b).

## 2. Background Knowledge

### 2.1 Description Logic and Uncertainty

The current work on including inductively derived information has focused on classification of assertions (ABox) in a Description Logic (DL) knowledge base, by associating uncertainty to its terminology (TBox). Description Logic provides constructors to build complex concepts and roles out of atomic ones [10], with various extensions derived to handle different types of constructs [17][10]. In recent years, much attention has been placed on the $\mathcal{SH}$ family of extensions, because it provides sufficient expressivity, useful for intended application domains. More recently, the $\mathcal{SHOQ}(\mathbf{D})$ extension has added the capability to specify qualified number restrictions, and the $\mathcal{SHOIN}(\mathbf{D})$ extension has combined singleton classes, inverse roles and unqualified number restrictions. Further, $\mathcal{SHOIN}(\mathbf{D})$ has been used to create the Web Ontology Language (OWL), which has been adopted as the web ontology standard by W3C [17]. OWL implements the open world assumption (OWA) [32] that if a statement is *unknown* it has not been falsified. In contrast, the closed world assumption (CWA) states that if a statement is not known to be true, it is false. These assumptions are related to defaults, which resolve ambiguities and missing values in a closed world system, benefits which cannot be assumed in the open world. New developments in inductive methods have been proposed to close the gap between CWA defaults and any ambiguities they introduce in the open world.

In the past several years, significant contributions have been made to introducing uncertainty to DL. Some notable ones have been the introduction of P-$\mathcal{SHOQ}$ $(\mathbf{D})$[15], a probabilistic extension to $\mathcal{SHOQ}$ $(\boldsymbol{D})$ [18][24], fuzzy $\mathcal{SHOIN}$ $(\mathbf{D})$ [26], a fuzzy extension to $\mathcal{SHOIN}$ $(\mathbf{D})$ [17], as well as BayesOWL [8] and PR-OWL [6], probabilistic extensions to OWL. These techniques offer new ways of querying, modeling, and reasoning with DL ontologies. P-$\mathcal{SHOQ}$ $(\mathbf{D})$ has provided a sound, complete, and decidable reasoning technique for probabilistic Description Logics. Fuzzy $\mathcal{SHOIN}$ $(\mathbf{D})$ demonstrates subsumption and

[3] The choice of the word *concept* is used in order to differentiate the general ontology *concept* and the lowest level of the use case *Felinae* ontology called *Class*, which will be identified by a capital letter and italics.

[4] We make a distinction between *matching* as the alignment between entities of different ontologies, and *mapping* as the directed version alignment of entities in one ontology to at most one entity in another, as in [11].

entailment relationship to hold to a certain degree, with the use of fuzzy modifiers, fuzzy concrete domain predicates, and fuzzy axioms. Fuzzy $\mathcal{SHOIN}(\mathbf{D})$ is an extension to work done on extending the $\mathcal{ALC}$ DL with fuzzy operators [27][28] (see Straccia et. al. [26] for a more complete list of extensions). BayesOWL converts an OWL TBox to a directed acyclic graph (DAG) with concept and relation nodes associated with Bayesian probabilities. PR-OWL is a language as well as a framework which allows ontologists to add probabilistic measures and reasoning to OWL ontologies. PR-OWL implements Multi-Entity Bayesian Networks (MEBN) [21], which extends axioms with Bayesian Network (BN) probabilities to first-order-logic (FOL) expressiveness. It should be noted that the key differences between probabilistic and fuzzy systems are that fuzzy uncertainty represents a degree of vagueness and lacks determinism [15], while probabilities represent dependencies and allow for deterministic reasoning.

A key task in probabilistic description logic is identifying which attributes to use, the relationships between them, and calculating the probabilities assigned to those relations. The goal is the capability of predicting the likelihood of corresponding attribute values. Various techniques have been applied to create probabilistic description logics. In [13], classification is performed by deriving a classification equation for non-linear models with the use of a support-vector-machine (SVM) classifier, with the optimal equation features, called kernel features, derived with genetic programming [7]. Rough Sets [25] have been applied to create a static probabilistic DL ontology [19], for the purpose of reasoning over data from different sources. In this work [19], rough fuzzy $\mathcal{SHOIN}(\mathbf{D})$ is introduced as an extension to fuzzy $\mathcal{SHOIN}(\mathbf{D})$. BayesOWL creates probabilities for OWL DLs by converting a DL to a DAG, and assigning probabilities to each edge using a conditional probability table (CPT), for two types of nodes; concept nodes and L-nodes (logical relations) [8]. As an example, the CPT probabilities for an *equivalent* L-node between $c_1$ and $c_2$, is *True*=1.0 if $[(c_1 \wedge c_2) \vee (\neg c_1 \wedge \neg c_2)]$=*True*, and T*rue*=0.0 otherwise, while a *complement* L-node is *True*=1.0 if $[(\neg c_1 \wedge c_2) \vee (\neg c_1 \wedge \neg c_2)]$=*True*, and *True*=0.0 otherwise.

### 2.2 Decision Trees

As a data structure, *decision trees* are used to represent the logical structures of classification rules for domain specific empirical data. The basic algorithm selects the attribute with the highest information gain for a particular class, and creates disjoint subsets based on that attribute's values. Ordinal attributes are split into two branches on the < and ≥ number restriction. For example the *size* attribute could be split to *large* and *small* classes based on the number of instances and their size values. Nominal attributes are treated as categorically disjoint sets, with as many branches as there are values. For example, the transitive relation, and more specifically enumerable instances of $\mathcal{SHOQ}$, would be able to express the ontology $\mathcal{O}_{\text{class}}$ relation $x\mathcal{R}y : [x \in \{\text{Country}\} \wedge y \in \{\text{France, Italy, Spain}\}]$. A DT classifying *Country* would be represented with a parent node *Country*, and three sub-nodes, *France*, *Italy*, and *Spain*. These could be further split on an ordinal attribute such as *population size* ranges, or another nominal attribute such as *language*. These subsets are smaller in cardinality, but more exact in precision in classifying a concept. The key factor in the classifying process is the attribute and value combinations which identify concepts best, which make up the classification rules. As mentioned in Section 1, the advantage in using an ontology is that this attribute/value factor is guided by the attribute's semantic relation to a particular concept. As described further in Section 3.2, this advantage is utilized in our algorithm to build DTs which select the most appropriate attributes and values which identify a semantic relationship deductively from the data.

### 2.3 Granular Computing

In section 2.1, we presented current work on introducing uncertainty to DL. As can be seen, it is beneficial to study the individual elements which make up a concept or cluster of concepts. It gives us a new understanding of what we viewed as atomic structures, and a new way of reasoning with them. This is the fundamental goal of granular computing [34], to view elements as parts of groups, and study the reasons why elements are grouped together by indistinguishability, similarity, proximity, and functionality [35].

**Definition 1 (Granule).** Granules are partitions of object space where objects are indistinguishable [19].

Any proposition which holds for a granule *Gr*, also holds for the complex concepts *Gr* is meant to identify, within a group of similar concepts. The benefits of using rough and fuzzy sets, is that they provide a level of granularity through inductive means, by defining crisp sets from fuzzy or possibilistic scoring models [30][19], and similar to DTs, are non-parametric [31]. The attributes used with granular boundaries are completely induced by the instances themselves. When viewed in the scope of ontologies, the notion of a granular ontology has been defined as "an inventory of entities existing in reality all of which belong to the same level of some granular partition" [2]. The authors argue that both the enduring entities such as substances, qualities, roles, and functions (SPAN), as well as perduring entities such as processes and their parts and aggregates (SNAP), are required in order to give a non-reductionism account of complex domains of reality. By inductively reducing the dimensionality of a concept, both rough sets and DTs are able to provide discrete partitions, required to identify and distinguish instances. Bitnner et al. [1] identifies the requirements for crisp and vague boundaries, which are provided by rough and fuzzy sets, respectively.

### 2.4 Ontology Matching

Ontology matching consists of matching a concept from one ontology to another. Several issues have been brought up as obstacles in the manual matching process [12][16], specifically inconsistency, incompletes and redundancy. This results in incorrectly defined relationships, missing information, or simply human error. Various techniques have been identified by Euzenat et al. [11], for automated and semi-automated matching techniques. Specifically *instance identification techniques,* such as comparing data values of instance data, are described to determine data correspondences, especially when ID keys are not available. When datasets are not similar to each other, disjoint *extension comparison techniques* are described, which can be based on statistical measures of class member features matched between entity sets [11]. The information created by our algorithm is targeted at datasets for such matchings. Random effects of DT classification algorithms can be stabilized using techniques such as bagging and stacking [33], where multiple trees are created and combined, and increase similarity measures of derived models. BayesOWL has been proposed to perform automatic ontology mapping [9] by associating probabilities with text based information, and using Jeffrey's Rule to propagate those probabilities. Text documents are classified using a classifier such Rainbow[5], and probabilities are assigned using the CPT process described in section 2.1. Tools such as OWL-CM [4] have begun looking at how similarity measures and uncertainties in the mapping process can be improved to improve access correspondences between text ontology entities.

### 2.5 Rule Insertion and Enhancement

Generating rules by inductive means allows us to add the axioms which govern an ontology. It would also be beneficial to enhance existing axioms, by introducing exceptions, and splitting axioms into two or more variations, to cover a broader scope of observations. To maintain a level of consistency, we require an increase in the granularity of the enhanced axiom, as it now covers a less broadly described observation. *Ripple down rules* (RDR) [20] allow us to add knowledge to existing axioms represented by a hierarchical structure, through such exceptions. This prolongs the usability and maintainability of existing rules, while they are refined and added to [20]. RDR exceptions can also introduce closed world defaults [20].

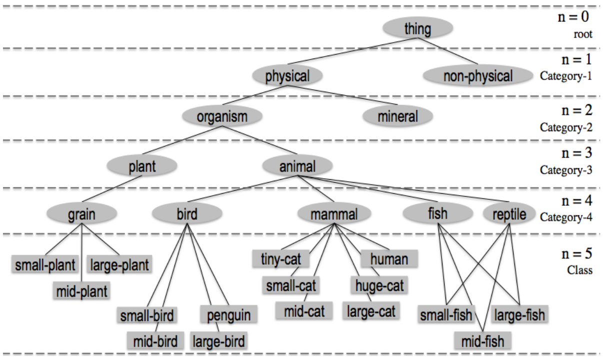


**Fig 2.** An ontology, split by levels *n*, which are used for iterating edges in our algorithm in section 3.2.

[5] http://www-2.cs.cmu.edu/~mccallum/bow/rainbow

## 3. Ontology Granulation

In this section, we describe our algorithm for adding granularity to ontologies, by using the decision trees induced from the data built to create the ontology. Our work differs from ODT [19] and SemTree [5], in that while they use an ontology to build a DT, we use DTs to add granules to an existing ontology. The deductively derived DTs will hold classification rules which may overlap with another set of rules for similar concepts in a different ontology. Our sample ontology is a small hierarchy of objects, with a breakdown on *physical* objects, and further broken down to *grains* and *animals*, as depicted in Figure 2. Target ontologies are ones which can be represented by a directed acyclic graph (DAG).

### 3.1 Database Preparation

Our algorithm uses supervised learning to build a decision tree model of the instances, the ontology $\mathcal{O}$ is trying to describe semantically. In order to apply the learning algorithm, $\mathcal{O}$ must first be represented in a format which can be used to perform classification. For that reason, instances which $\mathcal{O}$ describes are represented by a tuple, and for our purpose, we assume it is stored in a database $\mathcal{DB}$. For a relational database, multiple tables must be denormalized. In this process, all attributes and relationships are brought into a single table, with logical and hierarchical relations being represented as attributes in a single row. It is important to represent concepts at equivalent levels[6] by the same column $C_n$, with different classes as separate values[5]. This is depicted in Figure 2, with all nodes at level $n$=4, for example, representing possible values for the column *Category-4* = $C_4$ = {*bird, mammal, grain, fish, reptile*}. Table 1 demonstrates this hierarchy as a denormalized table with all other attributes. Multiple parent nodes are represented by a duplication of records with different category values, as illustrated by instances 10 to 14, being represented by a different parent in *Category-4*, *reptile* and *fish*, but the same *Class* value of *small-fish*.

**Definition 2 (Data preparation)**. Given the ontology $\mathcal{O}$ being granularized, the related database $\mathcal{DB}$ has

$f$ := number of attributes in normalized version of $\mathcal{DB}$

$a_i$ := attribute; $i = \{ 0, \dots ,f \}$

$$v_i := \begin{cases} \text{value of } a_i & \text{if } a_i \text{ is defined} \\ \text{null} & \text{otherwise} \end{cases}$$

$C_n$ := $a_i$ representing a concept group at level $n$; i.e. {*Category-1*, … , *Class*}

**Table 1.** Normalized Data Sample

| Instance # | country | length | width | height | weight | fly | walk | swim | move | growth | ID | Size | Category 1 | Category 2 | Category 3 | Category 4 | Class |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Algeria | 12 | 4 | 6 | 115 | N | Y | N | Y | Y | 63 | small | physical | organism | animal | mammal | small-cat |
| 2 | Amrcn-Samoa | 4 | 1 | 3 | 4 | N | Y | N | Y | Y | 353 | tiny | physical | organism | animal | mammal | tiny-cat |
| 3 | Armenia | 51 | 14 | 29 | 8282 | N | Y | ? | Y | Y | 354 | ? | physical | organism | animal | mammal | huge-cat |
| 4 | New-Zealand | 7 | 1 | 3 | 2 | Y | Y | N | Y | Y | 469 | small | physical | organism | animal | bird | small-bird |
| 5 | New-Zealand | 14 | 6 | 6 | 50 | Y | Y | N | ? | Y | 617 | ? | physical | organism | animal | bird | mid-bird |
| 6 | Åland-Islands | 17 | 10 | 17 | 289 | Y | ? | N | Y | Y | 767 | large | physical | organism | animal | bird | large-bird |
| 7 | Antarctia | 5 | 5 | 28 | 560 | N | Y | Y | Y | ? | 841 | ? | physical | organism | animal | bird | penguin |
| 8 | Antig&Brbda | 89 | 58 | 99 | 255519 | N | Y | N | Y | Y | 909 | mid | physical | organism | animal | mammal | human |
| 9 | Aruba | 75 | 55 | 43 | 88688 | N | Y | N | Y | Y | 912 | mid | physical | organism | animal | mammal | human |
| 10 | New-Zealand | 8 | 1 | 3 | 7.2 | N | N | Y | Y | Y | 1183 | small | physical | organism | animal | fish | small-fish |
| 11 | New-Zealand | 8 | 1 | 3 | 7.2 | N | N | Y | Y | Y | 1183 | small | physical | organism | animal | reptile | small-fish |
| 12 | New-Zealand | 7 | 1 | 4 | 8.4 | N | N | Y | Y | Y | 1185 | ? | physical | organism | animal | fish | small-fish |
| 13 | New-Zealand | 7 | 1 | 4 | 8.4 | N | N | Y | Y | Y | 1185 | ? | physical | organism | animal | fish | small-fish |
| 14 | New-Zealand | 7 | 1 | 4 | 8.4 | N | N | Y | Y | Y | 1186 | ? | physical | organism | animal | reptile | small-fish |
| 15 | Bahrain | 0.001 | 0.001 | 0.001 | 0.000 | ? | ? | ? | N | Y | 945 | small | physical | organism | plant | grain | small-plant |
| 16 | Anguilla | 1.001 | 0.001 | 3.001 | 0.000 | ? | ? | ? | N | Y | 1100 | mid | physical | organism | plant | grain | mid-plant |
| 17 | Bahamas | 4.000 | 3.000 | 10.00 | 1.200 | ? | ? | ? | N | Y | 1164 | ? | physical | organism | plant | grain | large-plant |

[6] It is not required for levels to align when matching concept signatures (see section 3.2) across ontologies, only when initially creating the DTs, since the parent-to-child concept classification is done in isolation from the rest of the tree.

### 3.2 Ontology Granulation Algorithm

The granulation process involves deriving rules which use ordinal number ranges and nominal category identifiers to classify specific ontology concepts. By identifying relationships between attributes in classifying an ontology concept, a class *signature*[7] may become apparent. This signature may later be used for ontology matching. We begin by listing elements needed to prepare the ontology for classification.

**Definition 3 (Ontology hierarchy).** A given ontology $\mathcal{O}$ has a hierarchical representation which contains

$\mathcal{O}_h$ := hierarchical representation of $\mathcal{O}$ (see Figure 2)
$levels(\mathcal{O}_h)$ := number of levels in $\mathcal{O}_h$
$n$ := $\{ 1, \dots, levels(\mathcal{O}_h) \}$; *where* $n = 0$ is the tree root
$c_{n_j}$ := concept $\in \mathcal{O}$ at level $n$; where $j = \{ 0, \dots, |C_n| \}$
$|c|$ := number of instances classified as $c$
$edge(c_{n_j}, c_{n\text{-}1_k})$ := edge between node $c_{n_j}$ and its parent node $c_{n\text{-}1_k}$

**Definition 4 (Attribute relevance).** The attributes chosen to build a decision tree to granularize $c_{n_j}$, depend on $rank(c_{n_j}, a_i)$, which is the relevance of $a_i$ in classifying $c_{n_j}$ and can be chosen by an expert or automatically through a ranking algorithm such as Definition 5.

Attributes of $\mathcal{DB}$, mainly, $A = \{ a_0, a_1, \dots, a_f \}$, are selected into the subset $A_n : A_n \subseteq A$, based on their ability to classify concepts at level $n$, and construct a DT. When constructing DTs, however, only attributes which are required to differentiate between DT models are included in the final tree. This subset $A_m : A_m \subseteq A_n$, is chosen to granularize $c_{n_j}$.

When choosing an attribute automatically based on its contribution to classification, various rankings can be used. The data mining tool we are using is an open source package called Weka [33], which provides several algorithms, such as information gain, entropy, and principal component. The information gain algorithm has produced the best results for our dataset.

**Definition 5 (Information gain)[8].** We evaluate the worth of an attribute by measuring the information gain with respect to the class. *InfoGain(Class, Attribute) = H(Class) - H(Class | Attribute).*

Our experience has indicated that choosing an attribute which is ranked significantly less than the attribute representing the parent node of $c_{n_j}$, Equation 2, will prevent choosing $a_i$ which resembles a parent node, and cause classification to suffer from over-fitting, producing less meaningful classification rules. In the same sense, attributes ranked closely to ones representing child nodes or which are close to 0 should be avoided, Equation 3, otherwise they will have a relatively high level of misclassification.

$$rank(a_i) \quad rank(c_{n\text{-}1_j}) \ . \tag{2}$$

$$0 \quad rank(c_{n+1_j}) \quad rank(a_i) \ . \tag{3}$$

**Definition 6 (Concept granulation).** Given the set $A_m$, attributes utilized by the DT, we use a classification algorithm[9] which produces several Bayesian models of the concept $c_{n_j}$, as leaf nodes of the DT. Each leaf node, which we call a *granule* $Gr$, produced

$\sigma$ = Bayesian probability of classifying $c_{n_j}$ correctly with a $Gr$.
$\varphi$ = coverage (number of instances in a $Gr$ classifying $c_{n_j}$) out of $|c|$.
$Pr$ = $\sigma\ (\varphi\ /\ |c|)$ : probability of $Gr$ being correct and its accuracy covering entire set of $c_n$ instances.

[7] By *signature*, we mean an identifying characteristic of the object being classified, and not a *signature* which describes non-logical symbols of a formal logic, or a *signature* in cryptography.

[8] Definition taken from the Weka 3.6.0 module *weka.attributeSelection.InfoGainAttributeEval*

[9] The Weka 3.6.0 module *weka.classifiers.trees.J48* contained good options for controlling the size of the tree, but the *weka.classifiers.trees.NBTree* module provided trees with the more useful Naïve Bayes classifiers at the leaf nodes.

where the *k-th granule* $Gr_k$ is comprised of a DT branch, producing an associated clause with

$$Op \in \{ \leq, >, = \}.$$

$$Pr_k\ Gr_k(c_{n_j}) \leftarrow (a_x\ Op_0\, v_x) \wedge (a_y\ Op_1\, v_y) \wedge \ldots \wedge (a_z\ Op_n\, v_z)\ .$$

The clause derived by the classification process uses values associated with the instances in the learning dataset. This places a dependency on all probabilities and the given value $v_i$ of each used attribute $a_i$ in the associated granule *Gr*. Any attribute not supplied with a value acts as a wild card and increases the probability (PR) of the associated granule Gr, while decreasing the accuracy. For probabilities to be meaningful, the number of instances of concepts should be approximately equal. This ensures each concept has equal representation in the DT. For example, if 95% of observations are of concept A and 5% of concept B, B will not be represented by the DT, as the probability of incorrectly choosing A is only 5%.

**Definition 7 (Concept signature).** Given a set of granules $Gr_k$ used to classify $c_{n_j}$, we create a clause with

$\Omega$ = Probability of $c_{n_j}$, calculated as sum of $c_n$ probabilities (*Pr*) with an associated coverage |*c*| .

$$\Omega_j\ Sig_j(c_{n_j}) \leftarrow (Pr_x Gr_x) \vee (Pr_y Gr_y) \vee \ldots \vee (Pr_z Gr_z)\ .$$

The basic algorithm, as described below, discovers a set of features *important* [10] to the identification and differentiation of a set of classes (steps 1 - 3). It then uses the features to build a DT (step 4), which results in a set of rules that identify the classes with different levels of coverage, accuracy, and probability. Each concept has an associated concept signature and probability (step 5). The derived rules are used to build the signature clause (step 8) and probability (step 9). The concept signature is then associated with *c* in the ontology hierarchy $\mathcal{O}_h$ (step 11).

***Granulation Algorithm***

```
1)   Denormalize $\mathcal{DB}$, applying ontology classes as attributes (see Section 3.1 for a
     discussion and Table 1 for an example).
2)   For each $n \in levels(\mathcal{O}_h)$
3)      Select attribute set $A_n$ using $rank(a_i)$, to best classify $C_n$, by combining:
           - Ontology author
           - Subject matter expert (SME)
           - Definition 4 and 5.
4)      Execute classification algorithm (Definition 6) to produce a DT classifying
        $C_n$, producing models in the form of conjunctions of ($a_i$ $Op$ $v_i$) as branches in
        the tree.
5)      Initialize $Sig_j$ and associated probability $\Omega_j$ for each $c_{n_j}$.
6)      For each $k \in z$; where z is the number of granules (leaf nodes) classifying c.
7)         Capture entire branch of a DT model for $c_{n_j}$, giving $Gr_k$ and associated $Pr_k$.
8)         Append $Gr_k(c_{n_j})$ to the $Sig(c_{n_j})$ clause with the OR operator.
9)         $\Omega_j = \Omega_j + Pr_k$ .
10)     End
11)     Associate $\Omega_j Sig(c_{n_j})$ to edge($c_{n_j}$, $c_{n-1_k}$ ) using ripple down rule (RDR).
12)  End
```

### 3.3 Matching Granules

The process of matching granules is comprised of 1) classifying an ontology node using $A_n$, 2) associating the derived signature $Sig_j$ with that concept's node, and finally 3) identifying characteristics in $Sig_j$ which resemble another signature $Sig_x$, of another ontology's concept. Guided by the edges in hierarchies of the individual ontologies (by associating classification targets with ontology nodes as in Figure 3), various combination of attributes reveal resembling patterns, as was demonstrated in Figure 1, and is expanded on in the use case in section 4.1. The implementation of the matching process is outside the scope of this paper, but we provide key ideas and issues which we have identified in section 5, and covered the state of the research in section 2.4. With successful granulation and concept matching, any existing *signatures* in

[10] *Importance* here is dependant on the application and available resources. We describe several possibilities in the following section.

the form of FOL rules, DL roles, or hierarchical DTs are attached to the edges or relations between concepts, possibly through the use of RDR.

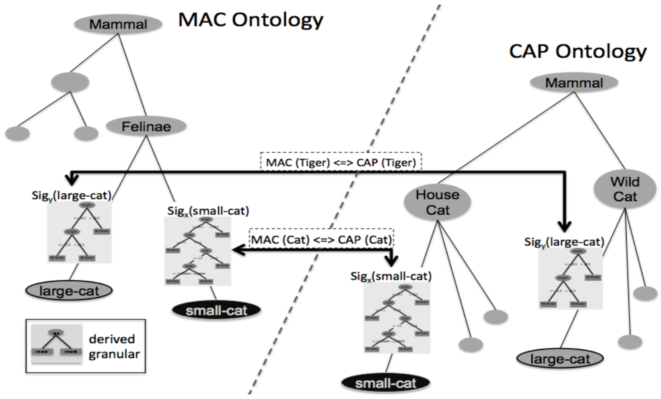


**Fig 3.** Concepts are mapped using derived *signatures* between two ontologies from section 4.1.1.

## 4. Motivating Example

### 4.1 Commerce Scenario

In a typical commerce use case, a manufacturer's goal is to find customers interested in purchasing their product. Our manufacturer Mats for Cats (MAC) has a set of criteria identifying the size and weight of cats, on which they base their product design. What they need now is a way to find a target market to advertise their product to. As part of the Semantic Web, the group Cats as Pets (CAP) has opened up their database and associated ontology of cat owners, with various types of *felinae*. CAP stores easily obtainable information about their cats, such as height, length, weight, colour, and location, and does not store a full ontology like the one stored by the Animal Diversity Web[11] (AWD) database. Also, because this is a world wide group, the pets range from house cats to large felines such as tigers. As a result, the stored information will vary, but correlation between attributes will classify various types of *felinae*. The MAC and CAP datasets are simulated, but suffer from real-world data issues such as incomplete and incorrect data, in addition to exhibiting features required for the matching process, to and test the attribute ranking and classification algorithms for their ability to handle such cases. Related data is required to map concepts, and the hypothesis is that even though perceptions may differ, the underlying occurrences will remain somewhat consistent [11]. Using the NBTree classifier in Weka, we classify *Felinae* as $F$ = {*tiny-cat*, *small-cat*, *mid-cat*, *large-cat, huge-cat*}, and derive the DT in Figure 4. Each leaf node represents a Bayesian model for each concept, with various degrees of probability $\sigma$ and coverage $\varphi$, and represent a single granule *Gr*. At this level, the decision is being made on *height*, *width*, *weight*, and *country*, but *country* was omitted by the DT, due to its low rank in its contribution to the classification.

*4.1.1 MAC Felinae Ontology Granulation*

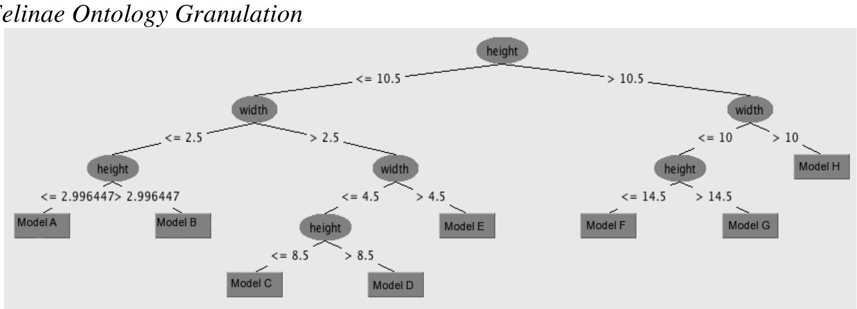


**Fig 4.** NBTree classifying MAC *Felinae* based on *height*, *width*, weight (omitted) and *country* (omitted).

[11] Animal Diversity Web: http://animaldiversity.ummz.umich.edu

**Table 2.** MAC granules build from the decision tree in Figure 2, using *height* ($x$) and *width* ($y$).

| Model | *Pr* | | Granule | |
|---|---|---|---|---|
| | $\sigma$ | $\varphi$ | | |
| A | 0.89 | 101 | Gr$_0$(tiny-cat) | ← ($x \leq 10.5$) ∧ ($y \leq 2.5$) ∧ ($x \leq 2.99$) |
| | 0.09 | 9 | Gr$_1$(small-cat) | ← ($x \leq 10.5$) ∧ ($y \leq 2.5$) ∧ ($x \leq 2.99$) |
| B | 0.92 | 44 | Gr$_2$(small-cat) | ← ($x \leq 10.5$) ∧ ($y > 2.5$) ∧ ($x > 2.99$) |
| C | 0.90 | 34 | Gr$_3$(small-cat) | ← ($x \leq 10.5$) ∧ ($y > 2.5$) ∧ ($y \leq 4.5$) ∧ ($x \leq 8.5$) |
| D | 0.58 | 13 | Gr$_4$(small-cat) | ← ($x \leq 10.5$) ∧ ($y > 2.5$) ∧ ($y \leq 4.5$) ∧ ($x > 8.5$) |
| | 0.29 | 6 | Gr$_5$(mid-cat) | ← ($x \leq 10.5$) ∧ ($y > 2.5$) ∧ ($y \leq 4.5$) ∧ ($x > 8.5$) |
| E | 0.64 | 6 | Gr$_6$(mid-cat) | ← ($x \leq 10.5$) ∧ ($y > 2.5$) ∧ ($y > 4.5$) |
| F | 0.87 | 26 | Gr$_7$(mid-cat) | ← ($x > 10.5$) ∧ ($y \leq 10$) ∧ ($y \leq 14.5$) |
| G | 0.78 | 93 | Gr$_8$(large-cat) | ← ($x > 10.5$) ∧ ($y \leq 10$) ∧ ($y > 14.5$) |
| H | 0.96 | 105 | Gr$_{10}$(huge-cat) | ← ($x > 10.5$) ∧ ($y > 10$) |

**Table 3.** MAC Signatures classifying *Felinae* built from granules in Table 2.

| $\Omega$ | | Signature | |
|---|---|---|---|
| $\Sigma\, Pr$ | $\lvert c\rvert$ | | |
| *0.89* | *101* | Sig$_0$(tiny-cat) | ← ($Pr_0\mathrm{Gr}_0$) |
| 0.78 | 100 | Sig$_1$(small-cat) | ← ($Pr_1Gr_1$) ∨ ($Pr_2Gr_2$) ∨ ($Pr_3Gr_3$) ∨ ($Pr_4Gr_4$) |
| *0.78* | *60* | Sig$_2$(mid-cat) | ← ($Pr_5\ Gr_5$) ∨ ($Pr_6Gr_6$) ∨ ($Pr_7Gr_7$) ∨ ($Pr_9Gr_9$) |
| *0.78* | *93* | Sig$_3$(large-cat) | ← ($Pr_8\ Gr_8$) |
| *0.96* | *105* | Sig$_4$(huge-cat) | ← ($Pr_{10}\ \mathrm{Gr}_{10}$) |

### *4.1.2 CAT Felinae Ontology Granulation*

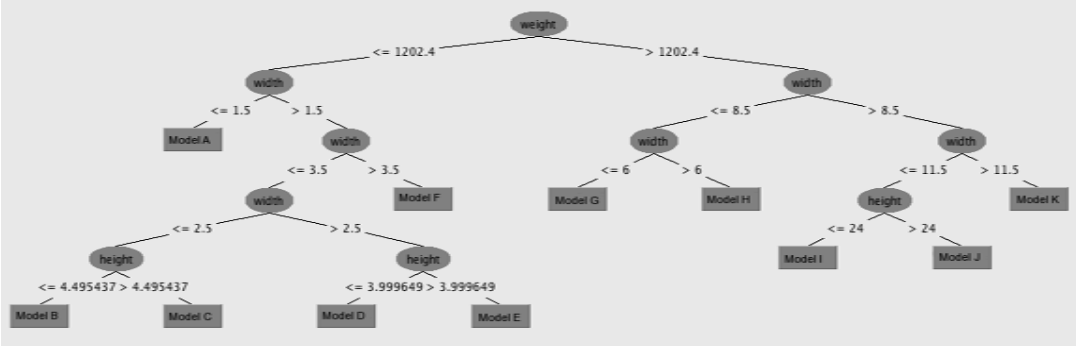


**Fig 5.** NBTree classifying CAP *Felinae* based on *height*, *width*, *weight*, (omitted) and *country* (omitted).

**Table 4.** CAP granules for *Felinae* classification based on height ($x$), width ($y$), weight ($z$).

| Model | *Pr* | | Granule | |
|---|---|---|---|---|
| | $\sigma$ | $\varphi$ | | |
| A | 0.51 | 24 | Gr$_0$(small-cat) | ← (weight ≤ 1202.4) ∧ ($y \leq 1.5$) |
| | 0.43 | 20 | Gr$_1$(mid-cat) | ← ($z \leq 1202.4$) ∧ ($y \leq 1.5$) |
| B | 0.09 | 4 | Gr$_2$(small-cat) | ← ($z \leq 1202.4$) ∧ ($y > 1.5$) ∧ ($y \leq 3.5$) ∧ ($y \leq 2.5$) ∧ ($x \leq 4.5$) |
| | 0.85 | 45 | Gr$_3$(tiny-cat) | ← ($z \leq 1202.4$) ∧ ($y > 1.5$) ∧ ($y \leq 3.5$) ∧ ($y \leq 2.5$) ∧ ($x \leq 4.5$) |
| C | 0.38 | 13 | Gr$_4$(small-cat) | ← ($z \leq 1202.4$) ∧ ($y > 1.5$) ∧ ($y \leq 3.5$) ∧ ($y \leq 2.5$) ∧ ($x > 4.5$) |
| | 0.54 | 19 | Gr$_5$(mid-cat) | ← ($z \leq 1202.4$) ∧ ($y > 1.5$) ∧ ($y \leq 3.5$) ∧ ($y \leq 2.5$) ∧ ($x > 4.5$) |
| D | 0.15 | 10 | Gr$_6$(small-cat) | ← ($z \leq 1202.4$) ∧ ($y > 1.5$) ∧ ($y \leq 3.5$) ∧ ($y > 2.5$) ∧ ($x \leq 4$) |
| | 0.80 | 56 | Gr$_7$(tiny-cat) | ← ($z \leq 1202.4$) ∧ ($y > 1.5$) ∧ ($y \leq 3.5$) ∧ ($y > 2.5$) ∧ ($x \leq 4$) |
| E | 0.40 | 15 | Gr$_8$(small-cat) | ← ($z \leq 1202.4$) ∧ ($y > 1.5$) ∧ ($y \leq 3.5$) ∧ ($y > 2.5$) ∧ ($x > 4$) |
| | 0.53 | 20 | Gr$_9$(mid-cat) | ← ($z \leq 1202.4$) ∧ ($y > 1.5$) ∧ ($y \leq 3.5$) ∧ ($y > 2.5$) ∧ ($x > 4$) |

| | | | | |
|---|---|---|---|---|
| F | 0.48 | 19 | $Gr_{10}$(small-cat) | ← $(z \leq 1202.4) \wedge (y > 1.5) \wedge (y > 3.5)]$ |
| | 0.45 | 18 | $Gr_{11}$(mid-cat) | ← $(z \leq 1202.4) \wedge (y > 1.5) \wedge (y > 3.5)$ |
| G | 0.67 | 7 | $Gr_{12}$(mid-cat) | ← $(z > 1202.4) \wedge (y \leq 8.5) \wedge (y \leq 6)$ |
| H | 0.87 | 26 | $Gr_{13}$(large-cat) | ← $(z > 1202.4) \wedge (y \leq 8.5) \wedge (y > 6)$ |
| I | 0.96 | 97 | $Gr_{14}$(large-cat) | ← $(z > 1202.4) \wedge (y > 8.5) \wedge (y \leq 11.5) \wedge (x \leq 24)$ |
| J | 0.95 | 78 | $Gr_{15}$(huge-cat) | ← $(z > 1202.4) \wedge (y > 8.5) \wedge (y \leq 11.5) \wedge (x > 24)$ |
| K | 0.87 | 26 | $Gr_{16}$(huge-cat) | ← $(z > 1202.4) \wedge (y > 8.5) \wedge (y > 11.5)$ |

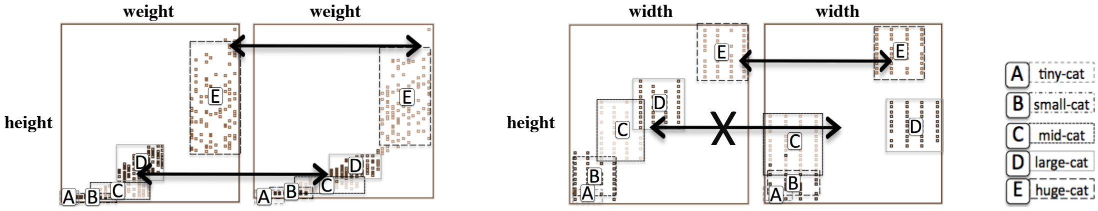
**Table 5.** CAP Signatures classifying *Felinae* built from granules in Table 4.

| *Ω* | | Signature | |
|---|---|---|---|
| *Σ Pr* | \|*c*\| | | |
| 0.82 | 101 | $Sig_0$(tiny-cat) | ← $(Pr_3Gr_3) \vee (Pr_7Gr_7)$ |
| 0.40 | 85 | $Sig_1$(small-cat) | ← $(Pr_0Gr_0) \vee (Pr_2Gr_2) \vee (Pr_4Gr_4) \vee (Pr_6Gr_6) \vee (Pr_8Gr_8) \vee (Pr_{10}Gr_{10})$ |
| 0.50 | 84 | $Sig_2$(mid-cat) | ← $(Pr_1Gr_1) \vee (Pr_5Gr_5) \vee (Pr_9Gr_9) \vee (Pr_{11}Gr_{11}) \vee (Pr_{12}Gr_{12})$ |
| 0.94 | 123 | $Sig_3$(large-cat) | ← $(Pr_{13}Gr_{13}) \vee (Pr_{14}Gr_{14})$ |
| 0.93 | 104 | $Sig_4$(huge-cat) | ← $(Pr_{15}Gr_{15}) \vee (Pr_{16}Gr_{16})$ |

### 4.2 Matching CAP and MAC Granules

For similar or equivalent domain databases, some attributes may demonstrate similarities, not only in individual attributes, but also in relation to another attribute in the database. The simplest measure is identifying similarities between each attribute and the concepts themselves. For example, the ranges of *width*, *height*, and *weight* values grouped by *Class*, may exhibit similarities between the MAC and CAP instances, showing a correlation between these two databases for the three attributes. A granule such as $Gr(mid\text{-}cat) \leftarrow (width > 0) \wedge (width \leq 4) \wedge (height > 4) \wedge (height \leq 8) \wedge (weight > 20) \wedge (weight \leq 50)$, could represent such clusters. A definition could be built by classifying a *Class* with a single attribute, like $Sig(mid\text{-}cat) \leftarrow ((width > 0) \wedge (width \leq 0.7)) \vee ((width > 1.1) \wedge (width \leq 2.1)) \vee ((width > 3.4) \wedge (width \leq 4.7))$.

Further, concentrating on the intersection of *weight* and *height*, we see a pattern of clusters, as depicted in Figure 6 (a) and (b). By representing these cluster graphs, we see overlapping clusters from (a) to (b), specifically cluster A (t*iny-cat*), B (*small-cat*), and E (*huge-cat*). In the centre of the graphs, we see two clusters C (*mid-cat*) and D (*large-cat*) overlapping each other to a lesser extent. We can begin to infer not only a matching between the *Classes* represented by these clusters (*tiny-cat*, *small-cat*, etc), but also between the attributes themselves (*height*, *weight*, etc).



(a) (b) (c) (d)

**Fig 6.** Attribute associations: *weight* × *height* for (a) MAC, (b) CAP; *height* × *width* for (c) MAC, (d) CAP.

Unfortunately, not all databases are this well aligned, and various measures of similarity must be considered. In Figure 6 (c) and (d), the correlation between the *height* and *width* attributes are analyzed, without a definite cluster correlation and overlapping as was observed in Figure 6 (a) and (b). As a result, a mix of similarities would need to be considered as a characteristics of a classification. As with Figure 6 (a) and (b), (c) and (d) contains a correlation between E (*huge-cat*) in the top-right, and the A (*tiny-cat*) and B (*small-cat*) clusters in the bottom-right. Unlike (a) and (b), however, no significant correlation exists between *mid-cat* and *large-cat*. A series of decision trees with various permutations of attributes would produce the best signature, such as a combination of both sets in Figure 6, for successful matching with another ontology's set of trees.

# 5. Conclusion

### 5.1 Discussion

In this paper, we present an algorithm for enhancing ontologies with inductively derived decision trees, in order to granulate the information being modeled by the ontology. The granulation process aims to produce partitions of characteristics of ontology concepts, based on the ontology's observed instances, such that the concepts are indistinguishable within those partitions, as per Definition 1. We then describe how these granules can be used to match concepts of different but similar ontologies with each other. We apply our algorithm to a simulated dataset of Felines, with a matching scenario in the commerce domain. The paper describes potential benefits of correlated data, which describes similar concepts, and how this relation can be utilized. The simulated database for MAC and CAP contained key real-life database features, positive and negative, required to demonstrate our algorithm.

### 5.2 Future Work

In our research, we have identified several key ontology matching observations and issues. It is important to find attributes in one ontology which are subsumed by a hybrid attribute derived from multiple attributes in the other. Relevant work has been done in the field of Object Based Representation Systems (OBRS) [3], where looking at subsumptions made about classified instances can lead to deducing new information about those instances. Our *granules* and *signatures* represent ranges and clusters which identify some class. For ordinal values, units of measure may be less relevant then ratios of values and their ranges, specifically when matching concepts at higher levels. For example, identifying traits in objects may depend on a correlation between two or more attributes. A long life span for one animal is short for another, so when grouping long life span factors, for example, it would make most sense to use the "relative" life span (in the form of ratios) of a particular species, when comparing life expectancy factors across multiple species.

Matching nominal attributes which may exist as sets (Colour(chair) = Red), attributes (chair.colour = Red) or properties (chair.Red) pose a challenge. In our preliminary research, the creation of a Boolean attribute in the normalized database for all possible sets or a value of an attribute or property, and assigning True or False to the values associated with a particular instance, the NBTree classification algorithm in Weka looked promising in identifying relationships between patterns of these values or sets. Properties such as Colour, which take on a single value, can be identified by recognizing a disjoint set amongst all instances, where a group of attributes (such as Red=True, Blue=False, Green=False, etc) which never have more then one True value for a group, can be a clue to a single set, attribute or property. During the matching process, any missing attributes would need to be inserted with default values of False. Further investigation is needed as this is a closed world assumption, which can be more harmful then beneficial.

**Acknowledgement:** Bart Gajderowicz gratefully acknowledges the discussions with Michael Grüninger of the University of Toronto, which benefited this research. The authors would like to thank Mikhail Soutchanksi of Ryerson University, and the numerous reviewers for their suggestions and comments.

# References

[1] Bittner, T., Smith, B. Granular Partitions and Vagueness, In Formal Ontology in Information Systems: Collected Papers from the Second International Conference, pp. 309-320 (2001)

[2] Bittner, T., Smith, B., Granular Spatio-Temporal Ontologies, In Proceedings of the AAAI Spring Symposium on Foundations and Applications of Spatio-temporal Reasoning (FASTR) (2003).

[3] Bisson, G., Why and How to Define a Similarity Measure for Object Based Representation Systems, In Towards Very Lare Knowledge Bases, pp. 236--246, IOS Press, Amsterdam (1995)

[4] Brodaric, B., Gahegan, M., Experiments to Examine the Situated Nature of Geoscientific Concepts. Spatial Cognition & Computation: An Interdisciplinary Journal, 7 (1), pp. 61--95 (2007)

[5] Bouza, A., Reif, G., Bernstein, A., Gall, H., SemTree: Ontology-Based Decision Tree Algorithm for Recommender Systems, In Proceedings of the 7th International Semantic Web Conference, Germany (2008)

[6] da Costa, P.C.G., Laskey, K.B., Laskey, K.J., PR-OWL: A Bayesian Ontology Language for the Semantic Web, Uncertainty Reasoning for the Semantic Web workshop (LNCS Vol.) pp. 88--107 (2008)

[7] De Jong, K.: Evolutionary computation: A unified approach. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation 2008, pp. 2245--2258 (2008)

[8] Ding, Z., Peng, Y., Pan, R., BayesOWL: Uncertainty modeling in semantic web ontologies, Studies in Fuzziness and Soft Computing, 204, pp. 3--29 (2006)

[9] Ding, Z., Peng, Y., Pan, R., Yu, Y., A bayesian methodology towards automatic ontology mapping, AAAI Workshop - Technical Report, WS-05-01, pp. 72--79 (2005)

[10] Erdur, Cenk, R., Seylan, Inanc, The design of a semantic web compatible content language for agent communication. Expert Systems , 25 (3), pp 268--294 (2008)

[11] Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, 67 illus., Hardcover, ISBN 978-3-540-49611-3, pp. 104--107 (2007)

[12] Falconer, S., Noy, N.F., Storey, M.A.: Ontology mapping - a user survey. In: Workshop on Ontology Matching (OM2007) at ISWC/ASWC2007, Busan, South Korea, November 2007, pp. 113--125 (2007)

[13] Fanizzi, N., d'Amato, C., Esposito, F., Statistical Learning for Inductive Query Answering on OWL Ontologies, In the proceedings of the International Semantic Web Conference, pp. 195--212 (2008)

[14] Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Sweetening WordNet with Dolce, In: AI Magazine, 24 (3), pp. 13-24 (2003)

[15] Giugno, R., Lukasiewicz, T., P-SHOQ(D): A Probabilistic Extension of SHOQ(D) for Probabilistic Ontologies in the SemanticWeb, Logics in Artificial Intelligence, 2424, pp. 86--97, Springer (2002)

[16] Gomez-Perez, A.: Handbook on Ontologies. In: International Handbooks on Information Systems, Springer, pp. 251--274 (2005)

[17] Horrocks, I., Patel-Schneider, P.F., van Harmelen, F., SHIQ and RDF to OWL: The Making of a Web Ontology, pp 7--26 (2003)

[18] Horrocks, I., Sattler., U., Ontology Reasoning In The SHOQ(**D**) Description Logic. In Proceedings of the 7th International Joint Conferences on Artificial Intelligence, pp 199--204 (2001)

[19] Klinov, P., Mazlack, L.J., Granulating Semantic Web Ontologies, In Proceedings of the 2006 IEEE International Conference on Granular Computing, pp. 431--434 (2006)

[20] Kwok, R., Translations of ripple down rules into logic formalisms, Proceedings of the Fourth Australian Knowledge Acquisition Workshop, The University of New South Wales, Sydney, Australia, pp. 44--56 (1999)

[21] Laskey, K.B., MEBN: A Language For First-Order Bayesian Knowledge Bases. Artificial Intelligence, 172(2-3), pp. 140-178 (2008)

[22] Matuszek, C., Cabrai, J., Witbrock, M., DeOliveira, J., An introduction to the syntax and content of Cyc, In: AAAI Spring Symposium - Technical Report, SS-06-05, pp. 44--49 (2006)

[23] Niles, I., Pease, A.: Towards a standard upper ontology. In: FOIS 2001. Proceedings of the international conference on Formal Ontology in Information Systems, Ogunquit, Maine, ACM Press, New York, NY, USA, pp. 2–9 (2001)

[24] Pan, J.Z., Horrocks., I., Semantic Web Ontology Reasoning In The SHOQ($\mathbf{D_n}$) Description Logic. In Proceedings of the Description Logic Workshop (2002)

[25] Pawlak, Z., Rough Sets, International Journal of Information and Computer Sciences, 11, pp. 341-356 (1982)

[26] Straccia, U., A Fuzzy Description Logic for the Semantic Web, In Fuzzy Logic And The Semantic Web, Capturing Intelligence, 4, pp. 167--181, Elsevier (2005)

[27] Straccia. U., A Fuzzy Description Logic. In Proceedings of the 15th National Conference on Artificial Intelligence, pp 594--599, Madison, USA (1998)

[28] Straccia, U., Reasoning Within Fuzzy Description Logics. Journal of Artificial Intelligence Research, 14, pp. 137--166 (2001)

[29] Zhang, J., Silvescu, A., Ontology-Driven Induction of Decision Trees at Multiple Levels of Abstraction. In: Proceedings of Symposium on Abstraction, Reformation, and Approximation. (2002)

[30] Yao, Y.Y., Granular Computing: Basic Issues and Possible Solutions, In Proceedings of the Joint Conference on Information Sciences, 5 (1), pp. 186-189 (2000)

[31] Sikder, I.U., Munakata, T., Application of Rough Set and Decision Tree for Characterization of Premonitory Factors of Low Seismic Activity, In Proceedings of Expert Systems with Applications, 36, 1, pp. 102--110 (2009)

[32] Stojanovic, L., Methods and tools for ontology evolution, Ph.D. Thesis, University of Karlsruhe, Germany (2004)

[33] Witten, IH., Frank, E., Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco (2005)

[34] Zadeh, L.A. Fuzzy Sets And Information Granularity, In Advances in Fuzzy Set Theory and Applications, Gupta, N., Ragade, R. and Yager, R. (Eds.), North-Holland, Amsterdam, pp. 3-18 (1979)

[35] Zadeh, L.A. Towards A Theory Of Fuzzy Information Granulation And Its Centrality In Human Reasoning And Fuzzy Logic, In Fuzzy Sets and Systems, 19, pp.111-127 (1997)

# Axiomatic First-Order Probability

Kathryn Blackmond Laskey

Department of Systems Engineering and Operations Research
George Mason University, Fairfax VA 22030, USA
klaskey@gmu.edu

**Abstract.** Most languages for the Semantic Web have their logical basis in some fragment of first-order logic. Thus, integrating first-order logic with probability is fundamental for representing and reasoning with uncertainty in the semantic web. Defining semantics for probability logics presents a dilemma: a logic that assigns a real-valued probability to any first-order sentence cannot be axiomatized and lacks a complete proof theory. This paper develops a first-order axiomatic theory of probability in which probability is formalized as a function mapping Gödel numbers to elements of a real closed field. The resulting logic is fully first-order and recursively axiomatizable, and therefore has a complete proof theory. This gives rise to a plausible reasoning logic with a number of desirable properties: the logic can represent arbitrarily fine-grained degrees of plausibility intermediate between proof and disproof; all mathematical and logical assumptions can be explicitly represented as finite computational structures accessible to automated reasoners; contradictions can be discovered in finite time; and the logic supports learning from observation.

**Keywords:** First-Order Logic, Probability.

## 1 Introduction

Logic-based languages have long been recognized as an effective means to represent information clearly, unambiguously, and in a manner that facilitates processing by machines. By far the most common logical basis for Semantic Web languages is classical first-order logic (FOL). This is no accident: its clear syntax, well-understood semantics, and complete proof theory make FOL a natural choice for computational knowledge representation and reasoning. However, FOL lacks a fundamental capability essential for semantically aware systems. As Jeffreys [1] put it, "Traditional or deductive logic admits only three attitudes to any proposition: definite proof, disproof, or blank ignorance." An intelligent reasoner must do more: it must assess the plausibility of uncertain hypotheses, make reasonable choices when the outcome is uncertain, and use observations to improve its representation of the world.

Probability is the unique plausible reasoning calculus that satisfies certain intuitively satisfying axioms of coherent reasoning (e.g., [2]). For this reason, probability has achieved a privileged status for plausible reasoning akin to FOL's privileged status with respect to logical reasoning. The past few decades have given rise to increasingly expressive probability-based languages, as well as a host of restricted languages designed for scalability. There is increasing interest in probability for semantic web applications [3].

It is often taken for granted that a new kind of logic is needed to capture essential aspects of plausible reasoning: "Ordinary logic seems to be inadequate by itself to cope with problems involving beliefs. In addition a theory of probability is required" [4]. Because of their built-in machinery for reasoning about functions, higher-order logic has been proposed as a natural logical basis for combining probability and logic [5]. On the other hand, its complete proof theory makes first-order logic attractive as a computational logic. Moreover, it is attractive to use the same logic to reason both about the domain itself and about the plausibility of statements about the domain. The question thus arises of whether an adequate formalization of probability is possible within first-order logic itself.

This paper formalizes, within standard first-order logic, a probabilistic logic powerful enough to express uncertainty about arbitrary first-order sentences. By formalizing probability as an axiomatic first-order theory, probabilities can be associated coherently with arbitrary first-order sentences, with no modification of traditional first-order semantics. To stay within axiomatic first-order logic, probabilities are defined not as real numbers, but as elements of a real closed field. An axiom schema is added to the standard axioms of the probability calculus to give "logical teeth" to the idea that probability zero events do not happen. The semantics proposed here connects naturally to algorithmic notions of randomness as proposed by Kolmogorov and Martin-Löf [6], [7], as well as to Dawid's [8] calibration criterion.

## 2 First-Order Probability

We begin with a first-order language $\mathcal{L}$ used to make assertions about a domain. $\mathcal{L}$ includes the usual logical symbols (variables, logical connectives, universal and existential quantifiers), together with a set of domain-specific predicate, function and constant symbols. Without loss of generality, $\mathcal{L}$ is taken to be a traditional, untyped first-order language.[1] To operationalize the requirement that assertions be expressible as a finite computational structure, a knowledge base (KB) is taken to be an axiomatic theory of $\mathcal{L}$. That is, a KB contains a consistent, recursive set $A$ of sentences of $\mathcal{L}$. The logical consequences of these axioms comprise a recursively enumerable set $T_A$ = Cn($A$), called the *theory* of $A$. Gödel's completeness theorem implies that $T_A$ is equal to the set $\{\sigma : A \vdash \sigma\}$ of sentences provable from $A$.

In general, a KB may be incomplete – it need not imply a definite truth-value for every sentence. In fact, a sufficiently powerful theory is necessarily incomplete. It is useful for a reasoner to grade the plausibility of propositions it can neither prove nor disprove. Probability is a natural candidate for this purpose.

It seems reasonable to define probability as a function mapping each sentence to a real number between zero and 1, in a manner that satisfies the standard identities of probability theory, and so that sentences provable (disprovable) from $A$ are assigned

[1] It is well known that a typed first-order logic can be re-expressed via a syntactic transformation as an untyped logic (cf., [7]).

probability 1 (0). This approach, natural as it seems, runs into difficulty. The first roadblock is that in standard first-order logic, arguments of functions must be elements of the domain, not sentences or propositions. The second roadblock is that the theory of the real numbers cannot be fully characterized as an axiomatic first-order theory. Several authors have shown that formalizing probabilities as a real numbers in the unit interval results in a theory that cannot be axiomatized, and that does not admit a complete proof theory (cf., [10], [11]). On the other hand, by abandoning the requirement that probabilities be real-valued, Bacchus [12] developed axiomatic probability logics that have a complete proof theory.

Given the mathematical impossibility of both an axiomatic first-order theory and a function mapping sentences to real numbers, which should be preferred? To answer this question, we step back to first principles, and consider fundamental requirements for a computational probabilistic logic. First, a computational logic should explicitly represent all mathematical and logical assumptions as finite computational structures accessible to an automated reasoner. Second, it should be possible for a reasoner to discover contradictions in a knowledge base, to identify when observations are inconsistent with a theory, and to prove any consequence entailed by a theory. Third, a logic for plausible reasoning must be able to associate measures of plausibility with propositions, to express degrees of plausibility intermediate between proof and disproof, and to do so in a logically coherent manner. All these requirements are met by the proposed formalism. Furthermore, the first two requirements are automatically satisfied if probability is formalized as a traditional first-order axiomatic theory, while the final requirement can be met without demanding either that probability be formalized as a function on sentences, or that probability values be real numbers. Hence, a first-order axiomatic theory is a fundamental requirement, whereas a function from sentences to real numbers is dispensable.

If probability is not a function mapping sentences to real numbers, then what is it? We formalize probability as a function mapping Gödel numbers to elements of a real-closed field (RCF). A RCF is the closest one can come to formalizing the real numbers within first-order logic. The real numbers are uniquely characterized up to isomorphism as an ordered field with the least upper bound property. The ordered field axioms formalize familiar properties of the real and rational numbers: addition, multiplication, additive and multiplicative inverses (hence, subtraction and division), distribution of multiplication over addition, and complete ordering. These axioms can be formalized fully in FOL. The defining property of the real numbers, that every bounded non-empty set of real numbers has a least upper bound, is not a first-order property. In a RCF, the least upper bound property holds for all definable relations. The RCF axioms are sufficient to characterize all first-order properties of the real numbers (cf., [13]). Thus, we assume probabilities are elements of a RCF.

Gödel showed that, given a sufficiently powerful formal system, domain elements (e.g., numbers) can be associated with sentences, formulas, and proofs. This device allows indirect expression of and reasoning about logical notions such as proof and consistency, while complying with FOL's prohibition against direct reference to sentences. Defining probabilities as a mapping from Gödel numbers to elements of a RCF allows us to develop a fully first-order axiomatization of probability. We argue later that our axioms capture the essential requirements for a computational logic of plausible reasoning.

## 3 The Probability Axioms

The original language $\mathscr{L}$ and axioms $A$ are augmented with additional symbols and axioms to provide the necessary logical apparatus for probabilistic reasoning. The augmented language and axioms are called $\mathscr{L}^*$ and $A^*$, respectively.

### 3.1 The Language

The language $\mathscr{L}^*$ has numerical constants 0 and 1; arithmetic ordering predicate $\leq$; arithmetic operators + and ×; one-place predicate symbols $\mathcal{R}$ and $\mathcal{N}$ to represent real and natural numbers; and the two-place function symbol $\mathcal{P}$ to represent probability. In addition, there is a predicate $\mathcal{D}$ to represent elements of the domain; and a countable collection $\mathcal{L}_1, \mathcal{L}_2, \ldots$ of labels as names for individuals. If $\mathscr{L}$ already has mathematical symbols and mathematical axioms consistent with our probability axioms, we can make use of the existing logical machinery; otherwise new symbols are added.

### 3.2 The axioms

*Domain axioms*. Our first step is to relativize the domain axioms to $\mathcal{D}$. This is achieved through a standard syntactic translation, e.g., $\forall x\ \varphi(x)$ becomes $\forall x\ \mathcal{D}(x) \rightarrow \varphi(x)$, and the sentence $\exists x\ \varphi(x)$ becomes $\exists x\ \mathcal{D}(x) \wedge \varphi(x)$ (c.f., [9]).

*Integer arithmetic*. We require enough integer arithmetic to allow Gödel numbering and reasoning about provability. The following axioms, together with the RCF axioms defined below (which apply to natural numbers by virtue of inclusion) serve this purpose:

N1. $\forall x\ \mathcal{N}(x) \rightarrow \mathcal{R}(x)$

N2. $\mathcal{N}(0)$

N3. $\forall x\ \mathcal{N}(x) \rightarrow \mathcal{N}(x+1)$

N4. $\forall x\ \forall y\ \mathcal{N}(x) \wedge \mathcal{N}(y) \rightarrow ((x < y+1) \rightarrow (x \leq y))$

N5. $\forall x\ \mathcal{N}(x) \rightarrow \neg(x < 0)$

N6. All universal closures of formulas $\mathcal{N}(x) \rightarrow (\varphi(0) \wedge \forall x(\varphi(x) \rightarrow \varphi(x+1))) \rightarrow \forall x\ \varphi(x)$, where $\varphi(x)$ has $x$ (and possibly other variables) free.

*Real closed field axioms*. As described above, probabilities are formalized as elements of a real closed field (RCF), the first-order theory of the real numbers. Axioms for a real closed field can be found in [13]. Note that by virtue of being natural numbers, the constants 0 and 1 are also real numbers.

R1. *Additive and multiplicative closure*: $\forall x\ \forall y\ \mathcal{R}(x) \wedge \mathcal{R}(y) \rightarrow \mathcal{R}(x+y) \wedge \mathcal{R}(x \cdot y)$

R2. *Commutativity*: $\forall x\ \forall y\ \mathcal{R}(x) \wedge \mathcal{R}(y) \rightarrow (x+y = y+x) \wedge (x \cdot y = y \cdot x)$

R3. *Associativity*: $\forall x \forall y \forall z\ \mathcal{R}(x) \wedge \mathcal{R}(y) \wedge \mathcal{R}(z) \rightarrow (((x+y) + z = x + (y + z)) \wedge ((x \cdot y) \cdot z = x \cdot (y \cdot z)))$

R4. *Identity*: $\mathcal{R}(0) \wedge \mathcal{R}(1) \wedge 0 \neq 1 \wedge (\ \forall x\ \mathcal{R}(x) \rightarrow ((x+0 = x) \wedge (x \cdot 1 = x))\ )$

R5. *Inverses*: $\forall x\ \mathcal{R}(x) \rightarrow (\ \exists y\ (x + y = 0) \wedge (x \neq 0 \rightarrow \exists y\ (xz = 1)\ )\ )$

R6. *Distributive Law*: $\forall x \forall y \forall z\ \mathcal{R}(x) \wedge \mathcal{R}(y) \wedge \mathcal{R}(z) \rightarrow (x \cdot (y + z) = (x \cdot y) + (x \cdot z))$

R7. *Total order*: $\forall x \forall y \forall z\ \mathcal{R}(x) \wedge \mathcal{R}(y) \wedge \mathcal{R}(z) \rightarrow ((x \leq y \vee y \leq x) \wedge (x \leq y \wedge y \leq x \rightarrow x{=}y) \wedge (x{\leq}y \wedge y{\leq}z \rightarrow x{\leq}z))$

R8. *Agreement of ordering with field operations*: : $\forall x \forall y \forall z\ \mathcal{R}(x) \wedge \mathcal{R}(y) \wedge \mathcal{R}(z) \rightarrow ((x{\leq}y \rightarrow x{+}z \leq y{+}z) \wedge ((0 \leq x \wedge 0 \leq y) \rightarrow 0 \leq x \cdot y))$

R9. *First-order closure*: The following axiom schema holds for all one-place formulas $\varphi(x)$: $\forall x\,(\varphi(x) \rightarrow \mathcal{R}(x)) \wedge \exists x\,\varphi(x) \wedge \exists y\,(\mathcal{R}(y) \wedge \forall x\,(\varphi(x) \rightarrow x \leq y)) \rightarrow \exists y\,(\mathcal{R}(y) \wedge \forall x\,(\varphi(x) \rightarrow x \leq y) \wedge \forall z\,(\mathcal{R}(z) \wedge \forall x\,(\varphi(x) \rightarrow x \leq z) \leftrightarrow y \leq z)$

Axiom schema R9 is the first-order "image" of the least upper bound axiom. It states that if $\varphi(x)$ represents a non-empty subset of the real numbers and $\varphi(x)$ has a real upper bound, then $\varphi(x)$ has a real least upper bound. Tarski [14] showed that the theory of real closed real fields can be characterized as an ordered field in which every element has a square root and every polynomial of odd degree has a root. R9 covers not only relations definable in the language of the real numbers, but also any real relation definable in $\mathcal{L}^*$. Thus, the above axioms are stronger than the standard RCF axioms.

*Probability axioms*. Good [4] stresses that probability is properly a two-place function P($E$|$H$), taken to mean the probability that would be assigned to the proposition $E$ if the proposition $H$ were known to be true. Good introduces the symbol $H^*$ to denote "the usual assumptions of logic and pure mathematics," which must be taken as given in all probability assessments. He makes no attempt to decide exactly what should be assumed as part of $H^*$, and says it is conceivable that $H^*$ cannot be expressed in a finite number of words. Because our concern is reasoning by computational agents, we depart from Good and insist that the underlying assumptions be formalized as a first-order axiomatic theory. We require that $H^*$ be expressed as a finite computational structure, with an effective procedure for generating the axioms explicitly, and an effective procedure for checking whether any given sentence is an axiom. In particular, we assume that $H^*$ includes the axioms N1-N5, R1-R9, and P1-P6 (below).

We introduce into $\mathcal{L}^*$ the two-place function symbol $\mathcal{P}$. The value of $\mathcal{P}$ represents a meaningful probability whenever the following conditions are met (*i*) the first argument of $\mathcal{P}$ is the Gödel number $\#\sigma$ of a sentence $\sigma$ of $\mathcal{L}^*$; (*ii*) the second argument of $\mathcal{P}$ is the Gödel number $\#\varphi(x)$ of a one-place open formula $\varphi(x)$ defining a relation representable in $T_{A^*} = \mathrm{Cn}(A^*)$; and (*iii*) the relation represented by $\varphi(x)$ contains the Gödel numbers of all axioms in $A^*$. The formula $\varphi(x)$ is used to represent the set of sentences whose Gödel numbers satisfy $\varphi$. $\mathcal{P}(\#\sigma, \#\varphi(x))$ represents the probability of $\sigma$, given that all sentences in the set represented by $\varphi(x)$ are true. Condition (*iii*) says that the domain axioms and probability axioms are taken as given. For readability, we write $\mathcal{P}(\sigma \mid \varphi)$ for $\mathcal{P}(\#\sigma, \#\varphi(x))$ and $\mathcal{P}(\sigma \mid \tau, \varphi)$ for $\mathcal{P}(\#\sigma, \#\psi(x))$, where $\#\psi(x)$ represents the union of the relation defined by $\varphi$ and $\{\#\tau\}$. That is, $\mathcal{P}(\sigma \mid \tau, \varphi)$ represents the likelihood of $\sigma$ under the assumption that $\tau$ and all sentences in the set represented by $\varphi(x)$ are true.

With this preamble, we now present the probability axioms. The axioms are stated informally for readability; stating them formally is straightforward. The axioms are universally quantified over (Gödel numbers of) sentences $\sigma$ and $\tau$, and formulas $\varphi$.

P1. $0 \leq \mathcal{P}(\sigma \mid \varphi) \leq 1$.

P2. If $A^* \vdash \sigma$, then $\mathcal{P}(\sigma \mid A^*) = 1$.

P3. If $\mathcal{P}(\sigma\wedge\tau \mid \varphi) = 0$, then $\mathcal{P}(\sigma\vee\tau \mid \varphi) = \mathcal{P}(\sigma \mid \varphi) + \mathcal{P}(\tau \mid \varphi)$.
P4. $\mathcal{P}(\sigma\wedge\tau \mid \varphi) = \mathcal{P}(\sigma \mid \tau, \varphi) \times \mathcal{P}(\tau \mid \varphi)$
P5. If $\sigma\leftrightarrow\tau$, then $\mathcal{P}(\sigma \mid \varphi) = \mathcal{P}(\tau \mid \varphi)$, and $\mathcal{P}(\gamma \mid \sigma, A^*) = \mathcal{P}(\gamma \mid \tau, A^*)$ for all $\gamma$.

The first three axioms are the usual axioms for finitely additive probability. P4 formalizes Bayesian conditioning. P5 is taken from Good [4], and formalizes the notion that logically equivalent propositions should be interchangeable with regard to rational degrees of belief.

Some authors (including Good and de Finetti) regard finite additivity as sufficient to formalize rational degrees of belief. Other authors consider countable additivity to be essential. Because countable additivity is typically taken for granted in applications, we regard it as essential. However, full formalization of countable additivity is not possible within FOL, because FOL cannot express the notion of an arbitrary infinite sequence of Gödel numbers. To formalize countable additivity, we adopt a condition introduced by Gaifman [15]. Gaifman's condition can be formalized as a first-order axiom schema. Informally, it is stated as:

P6. $\mathcal{P}(\forall x\ \psi(x) \mid \varphi)$ is the supremum of the values $\mathsf{P}(\psi(\kappa_1)\vee\cdots\vee\psi(\kappa_n) \mid \varphi)$, for all finite conjunctions $\psi(\kappa_1)\vee\cdots\vee\psi(\kappa_n)$ of sentences, formed by substituting constant terms of $\mathcal{L}^*$ into $\psi(x)$.

The constants $\kappa_i$ may be constants of the original language $\mathcal{L}$, numerical constants (0 or 1), or label constants (one of the $\mathcal{L}_i$). The label constants provide enough constants to cover individuals that might not otherwise be referenced explicitly.

### 3.3 Terminology

The following definitions provide some necessary terminology.

***Definition 1***: Let $\mathcal{L}$ be a first-order language. A $p$-language $\mathcal{L}^*$ for $\mathcal{L}$ is a language that augments $\mathcal{L}$ with symbols for domain elements, real and natural number arithmetic, and probability, as described in Section 3.1 above. A $p$-theory $T_{A^*}$ for an axiomatic theory $T_A$ in $\mathcal{L}$ augments the axioms $A$ of $T_A$ as described in Section 3.2 above, to include: (*i*) axioms relativizing axioms in $A$ to elements of the original domain; (*ii*) axioms N1-N5, R-1R9, and P1-P6; and (*iii*) additional axioms defining a domain-specific probabilistic theory. A $p$-theory $T_{A^*}$ containing only N1-N5, RCF, and P1-P6, with no domain axioms, is called the *base* $p$-theory for $\mathcal{L}$.

***Definition 2***: Let $\mathcal{L}$ be a first-order language; let $\mathcal{L}^*$ be a $p$-language for $\mathcal{L}$. An axiomatic theory $T_{A^*}$ of $\mathcal{L}^*$ is *probabilistically complete* if it assigns a unique probability $\mathcal{P}(\sigma \mid A^*)$ to every sentence $\sigma$ of $\mathcal{L}$. That is, $T_{A^*}$ is probabilistically complete if for every sentence $\sigma$ there is a unique real number $p_\sigma$ such that $T_{A^*} \vdash \mathcal{P}(\sigma \mid A^*) = p_\sigma$.

A probabilistically complete $p$-theory assigns a single RCF element to each sentence. Incomplete $p$-theories give rise to interval probabilities. Some writers have advocated founding the theory of probability on interval rather than point-valued probabilities (e.g., [4]). The possibility of incomplete $p$-theories is attractive when the KB designer is not able to specify a probability for every sentence. With the advent of first-order languages based on graphical probability models, it is now possible to define probabilistically complete $p$-theories suitable for many interesting problems, to develop workable knowledge engineering procedures for specifying $p$-theories, and to devise tractable inference and learning algorithms for $p$-theories.

***Definition 3*:** Let $\mathcal{L}$ be a first-order language; let $T$ be a theory of $\mathcal{L}$; and let $\mathcal{L}^*$ be a $p$-language for $\mathcal{L}$. An axiomatic theory $T_{A^*}$ of $\mathcal{L}^*$ *corresponds* to $T$ if $T \vdash \sigma$ implies $T_{A^*} \vdash \mathcal{P}(\sigma | A^*) = 1$ for any sentence $\sigma$ of $\mathcal{L}$. $T_{A^*}$ *corresponds strongly* to $T$ if $T \vdash \sigma$ if and only if $T_{A^*} \vdash \mathcal{P}(\sigma | A^*) = 1$.

Clearly, if the axioms $A$ of $T_A$ are included among the axioms $A^*$ of $T_{A^*}$, then $T_{A^*}$ corresponds to $T_A$. In general, augmenting $A$ with N1-N5, RCF, and P1-P6, will not determine a unique $p$-theory. If it is assumed that $A$ incorporates all objective, incontrovertibly true domain knowledge, then adding probabilistic axioms to complete a $p$-theory brings subjectivity into the KB.

Of course, in actual applications, it is rarely the case that all logical axioms are incontrovertibly true assertions. More realistically, some axioms will be highly questionable; others, though quite useful, may be downright false. Axioms in real KBs are carefully engineered to be "good enough for the task." A great deal of subjective judgment goes into developing a "good enough" KB. In short, the logical axioms of a KB are often as subjective as the probabilities, and sometimes more so.

***Definition 4*:** Let $\mathcal{L}$ be a first-order language, $\mathcal{L}^*$ a $p$-language for $\mathcal{L}$, and $T_{A^*}$ a probabilistically complete axiomatic theory of $\mathcal{L}^*$. Let $\varphi(x,y)$ be a formula of $\mathcal{L}^*$ that functionally represents a recursive sequence of Gödel numbers of sentences of $\mathcal{L}^*$ (i.e., for each natural number $n$, there is exactly one sentence $\sigma_n$ such that $\varphi(n, \#\sigma_n)$ is provable from $A^*$). We say the sequence $\sigma_1, \sigma_2, \ldots$ of sentences is *negligible* if for every RCF element $u > 0$ there is a natural number $n$ such that $\mathcal{P}(\sigma_1 \wedge \cdots \wedge \sigma_n | A^*) < u$. The sequence $\sigma_1, \sigma_2, \ldots$ is *certain* if for every RCF element $u > 0$ there is a natural number $n$ such that $\mathcal{P}(\sigma_1 \wedge \cdots \wedge \sigma_n | A^*) > 1 - u$.

A negligible sequence is vanishingly improbable. That is, the probabilities of its finite-length leading segments tend to zero as their lengths increase without bound. Clearly, any sequence containing a zero-probability sentence is negligible. We can define negligible or certain individual sentences or finite-length sequences of sentences in the obvious way, by appending infinitely many copies of a tautology to the end of the sequence. An individual sentence is certain if it has probability 1 and negligible if it has probability zero.

***Defintion 5*:** Let $\mathcal{L}$ be a first-order language; let $\mathcal{L}^*$ be a $p$-language for $\mathcal{L}$; let $T^*$ be a theory of $\mathcal{L}^*$. The *core* of $T^*$ is the set of sentences $\{\sigma : \sigma$ is certain under $T^*\}$.

## 4 Semantics

This section defines semantics for $p$-theories.

*Standard first-order semantics*. The logic set forth in this paper is a standard, untyped first-order logic. As such, it can be given standard first-order model-theoretic semantics.

A first-order structure for a theory in $\mathcal{L}^*$ is a pair $(D, m)$, where $D$ is a non-empty set called the domain of interpretation, and $m$ assigns to each function, constant, and relation symbol of $\mathcal{L}^*$ a function, constant or relation of the correct arity on $D$. A structure $(D, m)$ implies a truth-value for each sentence of $\mathcal{L}^*$. $(D, m)$ is called a *model* of $T^*$ if every sentence in $T^*$ is true in $(D, m)$. Models of $T^*$ are sometimes called *possible worlds* for $T^*$. A sentence $\sigma$ is implied by $A^*$ if it is true in all models of $A^*$, and satisfiable if it has a model.

Standard semantics for $p$-theories gives rise to a seeming conflict between logical truth and probabilistic certainty: there may be logically possible sentences that have probability zero. For example, we might represent successive tosses of a symmetric die as independent and identically distributed with probability 1/6 of landing on each face. In a hypothetical infinite sequence of tosses, the frequency of tosses that land on, say, the number 2 is certain to be 1/6. Any sequence of outcomes that does not have a limiting frequency of 1/6 is negligible, in the sense of D4. Nevertheless, *every* sequence of outcomes is logically possible. Standard first-order semantics cannot distinguish between typical realizations of this probabilistic process (i.e., "random looking" sequences with limiting frequency 1/6) and highly atypical realizations (e.g., sequences that have the incorrect limiting frequency, or exhibit some other unusual regularity, such as a 2 on every sixth toss). Suppose the sentence $\sigma_1$ asserts that the limiting frequency is 1/6 that the die lands on 2; the sentence $\sigma_2$ asserts that every toss comes up 2; and the sentence $\sigma_3$ asserts that a 2 occurs on the first two tosses. The sentence $\sigma_1$ has probability 1; $\sigma_2$ has probability zero; and $\sigma_3$ has probability 1/36. None of these sentences is either implied by or inconsistent with the logical axioms. Traditional first-order semantics seems to provide no way to differentiate among them. For this reason, many authors have considered standard first-order semantics inadequate for probabilistic theories, and have turned to alternative semantics.

*Measure models.* A common approach to giving semantics to probabilistic logics is through a probability measure on structures. Using results from measure theory, Gaifman [15] showed that a coherent probability assignment to quantifier-free sentences can be extended to a countably additive probability measure on a $\sigma$-algebra of subsets of $\{(D, m)\}$, the set of all structures on $D = \{\mathcal{L}_1, \mathcal{L}_2, \ldots\}$.[2] In measure model semantics, a sentence is assigned probability equal to the measure of the set of models of the sentence.

Just as traditional first-order semantics is defined in terms of set theory, measure model semantics is defined in terms of measure theory. Measure theory is the branch of real analysis used to formalize probability. In measure model semantics, it is generally taken for granted that probabilities are interpreted as real numbers. As noted above, the semantic condition that probabilities must be interpreted as real numbers results in a non-axiomatizable logic that lacks a complete proof theory. On the other hand, axiomatic set theory provides sufficient mathematical machinery to prove the standard results of measure theory. If $A^*$ contains set theory axioms, then the results of measure theory hold in all models of $A^*$. Therefore, a probabilistically complete $p$-theory that includes set theory axioms has a unique measure model. A probabilistically incomplete $p$-theory has a family of measure models, one for each

[2] Gaifman's domain of interpretation included the constants of the original language as well as the added constants; the original constants are then self-interpreted. This requires that no two constants be equal, an assumption we do not make.

distribution consistent with $T_{A*}$. Any measure model assigns probability zero to the set of models of negligible sequences.

Under measure model semantics, the logical and probabilistic aspects of a theory remain semantically distinct. Provable sentences are true in all ordinary models of $T_{A*}$, and hence have probability 1 in any measure model of $T_{A*}$. Unsatisfiable sentences are false in all models and hence have probability zero in any measure model. Other than P1-P6, probabilities for other sentences are unrestricted. In particular, a sentence may provably have probability 1 and yet be false in some models.

*Certainty restriction semantics*. If a sentence has probability 1, then conditioning on the sentence does not change its probability or the probability of any other sentence. Furthermore, because there are only countably many sentences, we can condition on *all* certain sentences – the core of the *p*-theory – without changing any probabilities. We can use this fact to rule out negligible sentences as models of a *p*-theory. Note that a *p*-theory $T_{A*}$ has enough logical machinery to define a provability predicate. We can thus introduce an axiom schema that infers $\sigma$ from $T_{A*} \vdash \mathcal{P}(\sigma \mid A^*) = 1$. Adding this axiom schema excludes provably negligible sentences as models of $T_{A*}$. We call this axiom schema the *certainty restriction*. Adding the certainty restriction schema to $T_{A*}$ reduces the set of models of $T_{A*}$ without changing either the probability of any sentence or any of the measure models consistent with $T_{A*}$.

A rational agent makes no practical distinction between propositions with probability one and those provable from its knowledge base. Many texts use limiting frequencies (as well as other certain events) to define the meaning of probability statements (e.g., that "fair die" means that the limiting frequency of tosses landing on each face is 1/6). This suggests that the certainty restriction captures some aspect of the intuitive semantics of probability as it is commonly applied and understood.

*Strong probabilistic semantics*. The certainty restriction and the conditioning restriction can be formulated in first-order logic. This means that these conditions can be imposed as satisfaction criteria for *p*-theories without any change to first-order semantics. However, these conditions cannot capture the stronger semantic notion that infinite-length negligible sequences should not occur in models of a probabilistic theory. Strong probabilistic semantics requires that no model of a *p*-theory may contain all sentences in a negligible sequence of sentences.

Each negligible sequence can be identified with an *effectively null* binary string, as defined by Martin-Löf [6]. Martin-Löf randomness has been studied extensively (c.f., [7]), and is popular as a characterization of what it means for a sequence to be a typical realization of a probability distribution on sequences. Dawid's [8] calibration criterion is closely related to Martin-Löf randomness: the set of uncalibrated sequences for a given probability distribution is effectively null for that distribution.

If the axioms $A^*$ of our *p*-theory are strong enough to formalize measure theory, then we can prove that the set of negligible sequences has probability zero under the measure model for $A^*$. Thus, negligible sequences can be excluded as models of $A^*$ without changing any probabilities. However, excluding the negligible sentences as models means abandoning traditional first-order semantics, because the proposition that a sequence is negligible cannot be formalized as a recursive first-order axiom schema. As a consequence, there is no complete proof system for probabilistic logic with strong probabilistic semantics.

*Frequency probability*. Some authors have argued that fundamentally different kinds of probability are required to formalize different metaphysical notions such as subjective degrees of belief, long-run frequencies, physical randomness, or algorithmic randomness. Others argue that a single kind of probability is adequate for all these metaphysical positions. Good ([4], [16]) and Barnett [17] discuss the different viewpoints on this issue.

Because the same mathematics is applied to reason about all these kinds of probability, and a proliferation of different logics complicates knowledge representation and knowledge interchange, it seems reasonable to investigate whether a single computational logic might be applicable to different notions of probability.

We have argued above that *p*-theories can represent subjective degrees of belief about propositions that can neither be proven nor disproven. We argue that *p*-theories can represent long-run frequencies and physical randomness. The basic idea derives from a theorem of de Finetti [18] stating that an infinitely exchangeable sequence of events is mathematically equivalent to one a frequentist or proponent of physical propensity would model as independent and identically distributed (iid) given an unknown parameter, together with a subjective probability distribution on values of the parameter. To the frequentist, the parameter corresponds to the unknown long-run frequency. To the propensity theorist, the parameter corresponds to the unknown propensity. To the strict subjectivist, the parameter is a modeling fiction that provides a parsimonious representation for an exchangeable sequence.

Infinitely exchangeable sequences in *p*-theories are represented as first-order axiom schemas stating that different orderings of finite-length initial segments of a sequence are equally probable. *P*-theories can also represent iid propositions with unknown parameters. Incomplete *p*-theories can be used if no subjective distribution is available (perhaps due to philosophical aversion to subjective probability) for the unknown parameter. Thus, frequency and propensity probability as well as degree of belief probability can be represented with *p*-theories.

*Summary: Semantics for p-theories*. With no change to standard first-order semantics, a complete *p*-theory assigns probabilities consistently to sentences of $\mathcal{L}^*$ such that the axioms $A^*$ have probability 1. By including the certainty restriction axiom schema in $A^*$, we can identify probability 1 with provability from $A^*$. If $A^*$ includes set theory axioms, there is a unique measure model (probability measure over models) for each complete *p*-theory. Requiring that all models of a *p*-theory be non-negligible in the sense of Martin-Löf would take us out of the realm of first-order model theory.

## 5 Learning and Dogmatism

An attractive feature of probability theory is its inbuilt support for learning from observation. We can add any new non-negligible axiom to a *p*-theory, and Bayesian conditioning can be used to obtain an updated *p*-theory with the evidence as an axiom. As new observations accrue, we obtain a sequence of *p*-theories, each containing additional axioms representing new information obtained since the previous *p*-theory in the sequence.

Fundamental to the scientific attitude is lack of dogmatism. In our context, non-dogmatism means assigning probability zero only to propositions known incontrovertibly to be false. A dogmatic theory will be overturned if it makes definite empirical predictions that turn out to be false. However, a theory can be dogmatic without ever being proven false. For example, if a theory starts out certain that a coin is fair, it can never learn that the coin is biased, no matter how many trials are observed. A theory that allows for bias will eventually become convinced that a biased coin is biased, even if it begins with a high likelihood that the coin is fair.

If we begin with an axiomatic theory $T_A$ of $\mathscr{L}$, and assume that the axioms $A$ represent a set of sentences known incontrovertibly to be true (whether by definition or by meticulous empirical observation), we would like to be able to represent a $p$-theory that assigns probability zero to exactly those sentences that can be disproven from $A$. We say such a $p$-theory corresponds *non-dogmatically* to $T_A$. There are many reasons, including tractability, convenience of specification, economy of communication, and the like, that we might choose to represent and reason with a dogmatic theory, as long as it is judged "good enough" for the task at hand. But as a matter of principle, we want the capability to represent a non-dogmatic theory, even if reasoning with it is impractical.

A result of Gaifman and Snir [11] would seem to doom any hope of finding a non-dogmatic theory. They proved that, under measure-model semantics, every axiomatizable theory is dogmatic. On the other hand, Laskey [19] described how to specify a non-dogmatic $p$-theory corresponding to any consistent, finitely axiomatizable first-order theory. This apparent inconsistency is resolved by noting that Gaifman and Snir assume that all true sentences of natural number arithmetic are base axioms of the logical language, which is therefore not axiomatizable. Gaifman and Snir's no-go result does not apply to axiomatic first-order probability logics. An inevitable price of the axiomatic first-order approach, implied by Tarski's undefinability theorem, is that any complete $p$-theory must assign probability intermediate between 0 and 1 to some sentences in the language of arithmetic. This is natural if we view probability as a degree of provability. No first-order axiom system can prove all true sentences of arithmetic; therefore, no axiomatic first-order probability logic can assign probability 1 to all true sentences of arithmetic.

## 6 Conclusion

As probabilistic languages find increasing application to the Semantic Web (e.g., [20]), there is a need for a logical foundation that integrates traditional SW logics with probability logic. The logic presented here formalizes probability as a standard axiomatic first-order theory, with no alteration to traditional first-order model theoretic semantics. Advantages of this approach are the ability to represent $p$-theories as finite computational structures amenable to machine processing, the availability of a complete proof system, and compatibility with semantics of traditional logic-based languages. While this paper focuses on expressive power of the logic, SW applications require tractability and scalability. Future work will consider tractable restrictions of the logic, as well as fast approximate inference methods.

## Acknowledgments

Grateful acknowledgement is extended to anonymous reviewers who provided helpful comments on previous drafts of some material contained in the present paper.

## References

1. Jeffreys, H. Theory of Probability. (3rd ed) Oxford University Press (1961)
2. Savage, L.J. The Foundations of Statistics. Wiley, New York (1954)
3. Costa, P. C. G., Laskey, K.B. and Lukasiewicz, T. Uncertainty Representation and Reasoning in the Semantic Web, in Semantic Web Engineering in the Knowledge Society, Idea Information Science (2008)
4. Good, I. J. Probability and the Weighing of Evidence. London: Charles Griffin and Co (1950)
5. Ng, K. S. and Lloyd, J. W. Probabilistic Reasoning in a Classical Logic. Journal of Applied Logic, vol. 7, no. 2, pp. 218-238 (2009).
6. Martin-Lof, P. The Definition of Random Sequences. Information and Control vol. 9, pp. 602-619 (1966)
7. Li, M. and Vitányi, P. An Introduction to Kolmogorov Complexity and Its Applications (2nd ed). Springer (1997)
8. Dawid, A. P. Calibration-Based Empirical Probability. The Annals of Statistics vol. 13, no. 4, pp. 1251-1274 (1985)
9. Enderton, H.B. A Mathematical Introduction to Logic: Harcourt Academic Press (2001)
10. Halpern, J.Y. An Analysis of First-Order Logics of Probability. Artificial Intelligence, vol. 46, pp. 311-50 (1991)
11. Gaifman, H. and M. Snir. Probabilities over Rich Languages, Journal of Symbolic Logic, vol. 47, pp. 495-548 (1982)
12. Bacchus, F. Representing and Reasoning with Probabilistic Knowledge: A Logical Approach to Probabilities. Boston, MA, MIT Press (1990)
13. Shoenfeld, J. R. Mathematical Logic. Association for Symbolic Logic (1967)
14. Tarski, A. A Decision Method for Elementary Algebra and Geometry. Univ. of California Press (1951)
15. Gaifman, H. Concerning Measures in First-Order Calculi. Israel Journal of Mathematics. vol. 2, pp. 1-18 (1964)
16. Good, I.J. Good Thinking: The Foundations of Probability and Its Applications. University of Minnesota Press (1983)
17. Barnett, V. Comparative Statistical Inference. (3rd ed.) Wiley, New York (1999)
18. de Finetti, Bruno. Theory of Probability: A Critical Introductory Treatment. Wiley, New York (1974, originally published in 1934).
19. Laskey, K.B. MEBN: A Language for First-Order Bayesian Knowledge Bases. Artificial Intelligence, vol. 172, no. 2-3, pp. 140-178 (2008)
20. Carvalho, R., Laskey, K.B., Costa, P., Ladeira, M., Santos, L., and Matsumoto, S. UnBBayes: Modeling Uncertainty for Plausible Reasoning in the Semantic Web. In: V. Kordic (ed.) Semantic Web, IN-TECH Publishing, ISBN: 978-953-7619-33-6 (in press)

# An Algorithm for Learning with Probabilistic Description Logics

José Eduardo Ochoa Luna and Fabio Gagliardi Cozman

Escola Politécnica, Universidade de São Paulo,
Av. Prof. Mello Morais 2231, São Paulo - SP, Brazil
eduardo.ol@gmail.com, fgcozman@usp.br

**Abstract.** Probabilistic Description Logics are the basis of ontologies in the Semantic Web. Knowledge representation and reasoning for these logics have been extensively explored in the last years; less attention has been paid to techniques that learn ontologies from data. In this paper we report on algorithms that learn probabilistic concepts and roles. We present an initial effort towards semi-automated learning using probabilistic methods. We combine ILP (Inductive Logic Programming) methods and a probabilistic classifier algorithm (search for candidate hypotheses is conducted by a Noisy-OR classifier). Preliminary results on a real world dataset are presented.

## 1 Introduction

Ontologies are key components of the Semantic Web, and among the formalisms proposed within Knowledge Engineering, the most popular ones at the moment are based on Description Logics (DLs) [1]. There are however relatively few ontologies available, and on very few subjects. Moreover, building an ontology from scratch can be a very burdensome and difficult task; very often two domain experts design rather different ontologies for the same domain [2]. Considerable effort is now invested into developing automated means for the acquisition of ontologies. Most of the currently pursued approaches do not use the expressive power of languages such as OWL, and are only capable of learning ontologies of restricted form, such as taxonomic hierarchies [3].

It is therefore natural to try to combine logic-based and probabilistic approaches to machine learning for automated ontology acquisition. Inspired by the success of Inductive Logic Programming (ILP) [4] and statistical machine learning, in this paper we describe methods that learn ontologies in the recently proposed Probabilistic Description Logic CR$\mathcal{ALC}$ [5]. In using statistical methods we wish to cope with the uncertainty that is inherent to real-world knowledge bases, where we commonly deal with biased, noisy or incomplete data. Moreover, many interesting research problems, such as ontology alignment and collective classification, require probabilistic inference over evidence.

Probabilistic Description Logics are closely related to Probabilistic Logics that have been extensively researched in the last decades [6, 7]. Some logics [8] admit inference based on linear programming, while other resort to independence assumptions and graph-theoretical models akin to Bayesian and Markov networks. Despite the potential applicability of Probabilistic Description Logics, learning ontologies expressed in these logics is a topic that has not received due attention. In principle, it might be argued that the same methods for learning probabilistic logics might be applied, with proper account of differences in syntax and semantics. In this paper we follow this path and report on some similarities and differences that may be of interest.

The semantics of the Probabilistic Description Logic $\textsc{cr}\mathcal{ALC}$ [5] is based on measures over interpretations and on assumptions of independence. Scalability issues for inference in $\textsc{cr}\mathcal{ALC}$ have been addressed so that we can run inference on medium size domains [9]. There are two learning tasks that deserve attention. First, learning *probability values*, perhaps through a maximum likelihood estimation method. We use this technique and, due to uncertainty in Semantic Web datasets, we employ the EM algorithm. The second task is learning *logical constructs*, where we are interested in finding a set of concepts and roles that best fit examples and where probabilistic assessments can be assigned. In ILP algorithms such as FOIL [10], one commonly relies on a cover function to evaluate candidate hypotheses. We approach learning concepts as a classification task, and based on an efficient probabilistic Noisy-OR classifier [11, 12], we guide the search among candidate structures.

Section 2 reviews key concepts useful to the remainder of the paper. In Section 3, algorithms for learning $\textsc{cr}\mathcal{ALC}$ ontologies are introduced. Once these algorithms have formally stated, we wish to explore semi-automated reasoning from a real world dataset — the Lattes curriculum platform. A first attempt at constructing a probabilistic ontology using this dataset is reported in Section 4.

## 2 Background

Assume we are provided with a repository of HTML pages where researchers and students have stored data about publications, courses, languages, and further relational data. In order to structure such knowledge we might choose to use ontologies. We may extract *concepts* such as Researcher and Person, and we may establish relationships such as $\sqsubseteq$ among them. These concepts are often expressed using Description Logics (Section 2.1). Suppose we are not able to precisely state membership relations among concepts, but instead we can give probabilistic assessments such as $P(\mathsf{Student}|\mathsf{Researcher}) = 0.4$. Such assessments are encoded in Probabilistic Description Logics such as $\textsc{cr}\mathcal{ALC}$ (Section 2.2). Suppose further that we look for automated means to learn ontologies given *assertions on concepts* such as Student(jane); this task is commonly tackled by Description Logic Learning algorithms (Section 2.3).

### 2.1 Description Logics

Description Logics (DLs) form a family of representation languages that are typically decidable fragments of First Order Logic (FOL) with particular semantics [13, 14]. DLs represent knowledge in terms of *objects*, *concepts*, and *roles.* Each *concept* in the set of concepts $N_{\mathsf{C}} = \{\mathsf{C}, \mathsf{D}, \ldots\}$ is interpreted as a subset of a domain (a set of objects). Each *role* in the set of roles $N_{\mathsf{R}} = \{\mathsf{r}, \mathsf{s}, \ldots\}$ is interpreted as a binary relation on the domain. Individuals represent the objects through names from the set of names $N_{\mathcal{I}} = \{\mathsf{a}, \mathsf{b}, \ldots\}$. Information is stored in a knowledge base divided in (at least) two parts: the TBox (terminology) and the ABox (assertions). The TBox describes the terminology by listing concepts and roles and their relationships. The ABox contains assertions about objects.

Complex concepts are built using atomic concepts, roles and constructors. Depending on the constructors involved one can obtain different expressive power and decidability properties. The semantics of a description is given by a *domain* $\Delta$ and an *interpretation*, that is a functor $\cdot^{\mathcal{I}}$. We refer to [13] for further background on Description Logics.

One of the central ideas in DL is *subsumption* [14]: Given two concepts descriptions $\mathsf{C}$ and $\mathsf{D}$ in $\mathcal{T}$, $\mathsf{C}$ subsumes $\mathsf{D}$ denoted by $\mathsf{C} \sqsupseteq \mathsf{D}$, iff for every interpretation $\mathcal{I}$ of $\mathcal{T}$ it holds that $\mathsf{C}^{\mathcal{I}} \supseteq \mathsf{D}^{\mathcal{I}}$. Also, $\mathsf{C} \equiv \mathsf{D}$ amounts to $\mathsf{C} \sqsupseteq \mathsf{D}$ and $\mathsf{D} \sqsupseteq \mathsf{C}$.

Subsumption is a useful inference mechanism that allow us to perform standard reasoning tasks such as *instance checking* and *concept retrieval.* Instance checking is valuable for our ILP methods because it amounts to produce class-membership assertions: $\mathcal{K} \models \mathsf{C}(\mathsf{a})$, where $\mathcal{K}$ is the knowledge base, $\mathsf{a}$ is an individual name and $\mathsf{C}$ is a concept definition given in terms of the concepts accounted for in $\mathcal{K}$.

### 2.2 The Logic CR$\mathcal{ALC}$

The logics mentioned in Section 2.1 do not handle uncertainty through probabilities. It might be interesting to assign probabilities to assertions, concepts, roles; the Probabilistic Description Logic CR$\mathcal{ALC}$ does just that.

CR$\mathcal{ALC}$ is a probabilistic extension of the DL $\mathcal{ALC}$. The following constructors are available in $\mathcal{ALC}$: *conjunction* $(\mathsf{C} \sqcap \mathsf{D})$, *disjunction* $\mathsf{C} \sqcup \mathsf{D}$, *negation* $(\neg\mathsf{C})$, *existential* restriction $(\exists r.\mathsf{C})$, and value restriction $(\forall r.\mathsf{C})$. Concepts inclusions and definitions are allowed and denoted by $\mathsf{C} \sqsubseteq \mathsf{D}$ and $\mathsf{C} \equiv \mathsf{D}$, where $\mathsf{C}$ is a concept name. The semantics is given by a domain $\mathcal{D}$ and an interpretation $\mathcal{I}$. A set of concept inclusions and definitions is a terminology. A terminology is acyclic if it is a set of concept inclusions/definitions such that no concept in the terminology uses itself.

A key concept in CR$\mathcal{ALC}$ is *probabilistic inclusion*, denoted by $P(\mathsf{C}|\mathsf{D}) = \alpha$, where $\mathsf{D}$ is a concept and $\mathsf{C}$ is a concept name. If the interpretation of $\mathsf{D}$ is the whole domain, then we simply write $P(\mathsf{C}) = \alpha$. We are interested in computing a *query* $P(\mathsf{A}_o(\mathsf{a}_0)|\mathcal{A})$ for an ABox $\mathcal{A} = \{\mathsf{A}_j(\mathsf{a}_j)\}_{j=1}^{M}$ (this is an *inference*). Assume also that $\mathsf{C}$ in role restrictions $\exists\mathsf{r}.\mathsf{C}$ and $\forall\mathsf{r}.\mathsf{C}$ is a concept name. As probabilistic inclusions must only have concept names in their conditioned concept, assessments such as $P(\forall\mathsf{r}.\mathsf{C}|\mathsf{D}) = \alpha$ or $P(\exists\mathsf{r}.\mathsf{C}|\mathsf{D}) = \alpha$ are not allowed.

We assume that every terminology is acyclic; this assumption allows one to draw any terminology $\mathcal{T}$ as a directed acyclic graph $\mathcal{G}(\mathcal{T})$: each concept name is a node, and if a concept $\mathsf{C}$ directly uses concept $\mathsf{D}$, then $\mathsf{D}$ is a *parent* of $\mathsf{C}$ in $\mathcal{G}(\mathcal{T})$. Each existential and value restriction is added to the graph $\mathcal{G}(\mathcal{T})$. As each one of these restrictions directly uses $\mathsf{r}$ and $\mathsf{C}$, the graph must contain a node for each role $\mathsf{r}$, and an edge from $\mathsf{r}$ to each restriction directly using it. Each restriction node is a *deterministic* node in that its value is completely determined by its parents.

The semantics of CR$\mathcal{ALC}$ is based on probability measures over the space of interpretations, for a fixed domain. Inferences can be computed by a first order loopy propagation algorithm that has been shown to produce good approximations for medium size domains [9].

### 2.3 Inductive Logic Programming and Description Logic Learning

ILP is a research field at the intersection of machine learning [15] and logic programming [16]. It aims at a formal framework as well as practical algorithms for inductively learning relational descriptions from examples and background knowledge. Learning is commonly regarded as a search problem [17]; indeed, in ILP there is a space of candidate solutions, the set of "well formed" hypotheses **H**, and an acceptance criterion characterizing solutions. In concept-learning and ILP the search space is typically structured by means of the dual notions of generalization and specialization.

A significant algorithm for ILP is FOIL [10]. This algorithm moves from an explicit representation of the target relation (as a set of tuples of a particular collection of constants) to a more general, functional definition that might be applied to different constants. For a particular target relation, FOIL finds clauses in FOL one at a time, removing tuples explained by the current clause before looking for the next through a *cover* method. FOIL uses an information-based heuristic to guide its search for simple, general clauses. Because of its simplicity and computational efficiency, we have chosen to develop a covered approach when learning Probabilistic Description Logics.

There have been notable efforts to learn ontologies in Description Logics; some of these previous results have directly inspired our work. As noted by Fanizzi et al [14], early work on learning in DLs essentially focused on demonstrating the PAC-learnability for various languages derived from CLASSIC. Many approaches to the problem of learning concept definitions in DL formalisms can be classified in two categories [2]: in one category the problem is approached by translating it to another formalism in which concept learning has already been investigated, while in the other category the problem is approached in the original formalism.

One example of the first approach can be found in the work of Kietz [18], where the hybrid language CARIN-$\mathcal{ALN}$ is used [19]. This language combines a complete structural subsumption service in a DL with Horn logic, where terms are individuals in the domain. Likewise, in the $\mathcal{AL}$-log framework [20], DATALOG clauses are combined with $\mathcal{ALC}$ constructs. In the same direction, $\mathcal{DL}$+log [21] allows for the tight integration of DLs and DATALOG. Arguably, the decidable knowledge representation framework $\mathcal{SHIQ}$+log [1], is the most powerful among the ones currently available for the integration of DLs and clausal logic. First, it relies on the very expressive DL $\mathcal{SHIQ}$. Second, it allows for inducing a definition for DL concepts thus having ontology elements not only as input but also as output of the learning process.

The problem of translation turns out to be similar to an ILP problem. There are two issues to address: incompatibilities between DLs and Horn Logic, and the fact that the OWA[1] is used in DLs.

The other approach, solving the learning problem in the original formalism, can be found in the work of Cohen and Hirsh [22], which uses a pure DL-based approach for concept learning, in this case on the CLASSIC DL language. In these algorithms, ILP has been a significant influence, as refinement operators have been extensively explored. Badea and Nienhuys-Cheng [23] suggest a refinement operator for the $\mathcal{ALER}$ description logic. They also investigate some theoretical properties of refinement operators that favour the use of a downward refinement operator to enable a top-down search.

Learning algorithms for DLs (in particular for the language $\mathcal{ALC}$) were created by Iannone et al [2] that also make use of refinement operators. Instead of using the classical approach of combining refinement operators with a search heuristic, they developed an example driven learning method. The language, called YINYANG, requires lifting the instances to the concept level through a suitable approximate operator (most specific concepts MSCs) and then start learning from such extremely specific concept descriptions. A problem of these algorithms is that they tend to produce unnecessarily long concepts. One reason is that MSCs for $\mathcal{ALC}$ and more expressive languages do not exist and hence can only be approximated.

These disadvantages have been partly mitigated in the work of Lehmann [24], where approximations are not needed because it is essentially based on a genetic programming procedure lying on refinement operators whose fitness is computed on the grounds of the covered instances. In the DL-LEARNER system [3] further refinement operators and heuristics have been developed for the $\mathcal{ALC}$ logic.

The DL-FOIL system [14] is a new DL version of the FOIL [10] algorithm, that is adapted to learning the DL representations supporting the OWL-DL language. The main components of this new system are represented by a set of refinement operators borrowed from other similar systems [2, 3] and by a different gain function (proposed in FOIL-I [25]) which must take into account the OWA inherent to DLs. In DL-FOIL, like in the original FOIL algorithm, the generalization routine computes (partial) generalizations as long as they do not cover any negative example. If this occurs, the specialization routine is invoked for solving these sub-problems. This routine applies the idea of specializing using the (incomplete) refinement operator. The specialization continues until no negative example is covered (or a limited number of them).

[1] Open World Assumption mantains that an object that cannot be proved to belong to a certain concept is not necessarily a counterexample for that concept [14].

## 3 Probabilistic Description Logic Learning

In this section, we focus on learning DL axioms and probabilities tailored to CR$\mathcal{ALC}$. To learn the terminology component we are inspired by Probabilistic ILP methods and thus we follow generic syntax and semantics given in [16]. The generic supervised concept learning task is devoted to finding axioms that best represent assertions positive (covered) and negatives, in a probabilistic setting this *cover relation* is given by:

**Definition 1. (Probabilistic Covers Relation)** *A probabilistic covers relation takes as arguments an example $e$, a hypothesis $H$ and possibly the background theory $B$, and returns the probability value $P(e|H, B)$ between 0 and 1 of the example $e$ given $H$ and $B$, i.e., $covers(e, H, B) = P(e|H, B)$.*

Given Definition 1 we can define the Probabilistic DL learning problem as follows [16]:

**Definition 2. (The Probabilistic DL Learning Problem)** *Given a set $E = E_p \cup E_i$ of observed and unobserved examples $E_p$ and $E_i$ (with $E_p \cap E_i = \emptyset$) over the language $\mathcal{L}_E$, a probabilistic covers relation $covers(e, H, B) = P(e|H, B)$, a logical language $\mathcal{L}_H$ for hypotheses, and a background theory $B$, find a hypothesis $H^*$ such that $H^* = \arg\max_H score(E, H, B)$ and the following constraints hold: $\forall e_p \in E_p : covers(e_p, H^*, B) > 0$ and $\forall e_i \in E_i : covers(e_i, H^*, B) = 0$. The score is some objective function, usually involving the probabilistic covers relation of the observed examples such as the observed likelihood $\prod_{e_p \in E_p} covers(e_p, H^*, B)$ or some penalized variant thereof.*

Negative examples conflict with the usual view on learning examples in statistical learning. Therefore, when we speak of positive and negative examples we are referring to observed and observed ones.

As we focus in CR$\mathcal{ALC}$, $B = \mathcal{K} = (\mathcal{T}, \mathcal{A})$, and given a target concept $\mathsf{C}$, $E = Ind^+_{\mathsf{C}}(\mathcal{A}) \cup Ind^-_{\mathsf{C}}(\mathcal{A}) \sqsupseteq Ind(\mathcal{A})$, are positive and negative examples or individuals. For instance, candidate hypotheses can be given by $\mathsf{C} \sqsupseteq H_1, \ldots, H_k$, where $H_1 = \mathsf{B} \sqcap \exists \mathsf{D}.\top, H_2 = \mathsf{A} \sqcup \mathsf{E}, \ldots$.

We assume each candidate hypothesis together with the example $e$ for the target concept as being a probabilistic variable or feature in a probabilistic model[2]; according to available examples, each candidate hypothesis turns out to be true, false or unknown whether result for instance checking $\mathsf{C}(\mathsf{a})$ on $\mathcal{K}, Ind(\mathcal{A})$ is respectively true, false or unknown. The learning task is restricted to finding a probabilistic classifier for the target concept.

A suitable framework for this probabilistic setting is the Noisy-OR classifier, a probabilistic model within the Bayesian networks classifiers commonly referred to as models of independence of clausal independence (ICI) [12]. In a Noisy-OR classifier we aim at learning a class $C$ given a large number of attributes.

As a rule, in an ICI classifier for each attribute variable $A_j, j = 1, \ldots, k$ ($\mathbf{A}$ denotes the multidimensional variable $(A_1, \ldots, A_k)$ and $\mathbf{a} = (a_1, \ldots, a_k)$

[2] A similar assumption is adopted in the nFOIL algorithm [26].

its states) we have one child $A'_j$ that has assigned a conditional probability distribution $P(A'_j|A_j)$. Variables $A'_j, j = 1, \ldots, k$ are parents of probability of the class variable $C$. $P_M(C|\mathbf{A}')$ represents a deterministic function $f$ that assigns to each combination of values $(a'_1, \ldots, a'_k)$ a class $c$. A generic ICI classifier is illustrated in Figure 1 .

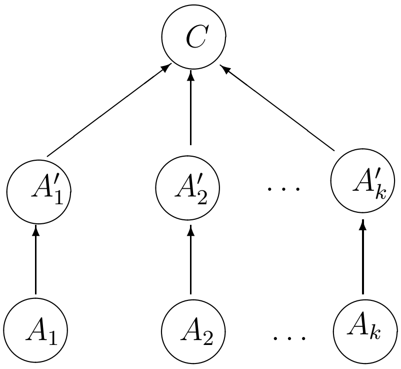


**Fig. 1.** ICI models [12].

The probability distribution of this model is given by [12]:

$$P_M(c, \mathbf{a}', \mathbf{a}) = P_M(c|\mathbf{a}') \prod_{j=1}^{k} P_M(a'_j|a_j) \cdot P_M(a_j),$$

where the conditional probability $P_M(c|\mathbf{a}')$ is one if $c = f(\mathbf{a}')$ and zero otherwise. The Noisy-OR model is an ICI model where $f$ is the OR function:

$$P_M(C = 0|\mathbf{A}' = 0) = 1 \text{ and } P_M(C = 0|\mathbf{A}' \neq 0) = 0.$$

The joint probability distribution of the Noisy-OR model is

$$P_M(\cdot) = P_M(C|A'_1, \ldots, A'_k) \cdot \left(\prod_{j=1}^{k} P_M(A'_j|A_j) \cdot P_M(A_j)\right).$$

It follows that

$$P_M(C = 0|\mathbf{A} = \mathbf{a}) = \prod_{j} P_M(A'_j = 0|A_j = a_j), \quad (1)$$

$$P_M(C = 1|\mathbf{A} = \mathbf{a}) = 1 - \prod_{j} P_M(A'_j = 0|A_j = a_j). \quad (2)$$

Using a threshold $0 \leq t \leq 1$ all data vectors $\mathbf{a} = (a_1 \ldots, a_k)$ such that $P_M(C = 0|\mathbf{A} = \mathbf{a}) < t$ are classified to class $C = 1$.

The Noisy-OR classifier has the following semantics. If an attribute $A_j$ is in a state $a_j$ then the instance $(a_1, \ldots, a_j, \ldots, a_k)$ is classified as $C = 1$ unless there is an inhibitory effect, with probability $P_M(A'_j = 0|A_j = a_j)$. All inhibitory effects are assumed to be independent. Therefore the probability that an instance does not belong to class $C$ ($C = 0$), is a product of all inhibitory effects $\prod_j P_M(A'_j = 0|A_j = a_j)$. For learning this classifier the EM-algorithm has been proposed [12]. The algorithm is directly applicable to any ICI model; in fact, an efficient implementation resort to a transformation of an ICI model using a hidden variable (further details in [12]). We now shortly review the EM-algorithm tailored to Noisy-OR combination functions.

Every iteration of the EM-algorithm consists of two steps: the expectation step (E-step) and maximization step (M-step). In a transformed decomposable model the E-step corresponds to computing the expected marginal count $n(A'_l, A_l)$ given data $D = \{\mathbf{e}^1, \ldots, \mathbf{e}^n\}$ ($\mathbf{e}^i = \{c^i, \mathbf{a}^i\} = \{c^i, a^i_1, \ldots, a^i_k\}$) and model M:

$$n(A'_l, A_l) = \sum_{i=1}^{n} P_M(A'_l, A_l|\mathbf{e}^i) \text{ for all } l = 1, \ldots, k$$

where for each $(a'_l, a_l)$

$$P_M(A'_l = a'_l, A_l = a_l|\mathbf{e}^i) = \begin{cases} P_M(A'_l = a'_l|\mathbf{e}^i) & \text{if } a_l = a^i_l, \\ 0 & \text{otherwise.} \end{cases}$$

Assume a Noisy-Or classifier $P_M$ and an evidence $C = c, \mathbf{A} = \mathbf{a}$. The updated probabilities (the E-step) of $A'_l$ for $l = 1, \ldots, k$ can be computed as follows [12]:

$$P_M(A'_l = a'_l|C = c, \mathbf{a})$$

$$= \begin{cases} 1 & \text{if } c = 0 \text{ and } a'_l = 0, \\ 0 & \text{if } c = 0 \text{ and } a'_l = 1, \\ \frac{1}{z} \cdot \left( P_M(A'_l = 0|A_l = a_l) - \prod_j P_M(A'_j = 0|A_j = a_j) \right) & \text{if } c = 1 \text{ and } a'_l = 0, \\ \frac{1}{z} \cdot P_M(A'_l = 1|A_l = a_l) & \text{if } c = 1 \text{ and } a'_l = 1, \end{cases}$$

where $z$ is a normalization constant. The maximization step corresponds to setting

$$P^*_M(A'_l|A_l) = \frac{n(A'_l, A_l)}{n(A_l)}, \text{ for all } l = 1 \ldots, k.$$

Given the Noisy-OR classifier, the complete learning algorithm is described in Figure 2, where $\lambda$ denotes the maximum likelihood parameters. We have used the refinement operators introduced in [3] and the Pellet reasoner[3] for instance checking. It may happen that during learning a given example for a candidate hypothesis $H_i$ cannot be proved to belong to the target concept. This is not necessarily a counterexample for that concept. In this case, we can make use of

[3] http://clarkparsia.com/pellet/.

**Input**: a target concept $\mathsf{C}$, background knowledge $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a training set $E = Ind^{+}_{\mathsf{C}}(\mathcal{A}) \cup Ind^{-}_{\mathsf{C}}(\mathcal{A}) \subseteq Ind(\mathcal{A})$ containing assertions on concept $\mathsf{C}$.
**Output**: induced concept definition $\mathsf{C}$.

```
Repeat
    Initialize $\mathsf{C}' = \bot$
    Compute hypotheses $\mathsf{C}' \sqsupseteq H_1, \ldots, H_n$ based on refinement operators for $\mathcal{ALC}$
      logic
    Let $h_1, \ldots, h_n$ be features of the probabilistic Noisy-OR classifier, apply the EM
      algorithm
    For all $h_i$
        Compute score $\prod_{e_p \in E_p} covers(e_p, h_i, B)$
    Let $h'$ the hypothesis with the best score
According to $h'$ add $H'$ to $\mathsf{C}$
Until $score(\{h_1, \ldots, h_i\}, \lambda_i, E) > score(\{h_1, \ldots, h_{i+1}\}, \lambda_{i+1}, E)$
```

**Fig. 2.** Complete learning algorithm.

the EM algorithm of the Noisy-OR classifier to estimate the class ascribed to the instance.

In order to learn probabilities associated to terminologies obtained for the former algorithm we commonly resort to the EM algorithm. In this sense, we are influenced in several respects from approaches given in [16].

## 4 Preliminary Results

To demonstrate feasibility of our proposal, we have run preliminary tests on relational data extracted from the Lattes curriculum platform, the Brazilian government scientific repository[4]. The Lattes platform is a public source of relational data about scientific research, containing data on several thousand researchers and students. Because the available format is encoded in HTML, we have implemented a semi-automated procedure to extract content. A restricted database has been constructed based on randomly selected documents. We have performed learning of axioms based on elicited asserted concepts and roles, further probabilistic inclusions have been added according to the CR$\mathcal{ALC}$ syntax. Figure 3 illustrates the network generated for a domain of size 2.

For instance, to properly identify a professor, the following concept description has been learned:

$$\begin{aligned}\mathsf{Professor} \equiv\ & \mathsf{Person} \\ & \sqcap(\exists \mathsf{hasPublication.Publication} \sqcup \exists \mathsf{advises.Person} \sqcup \exists \mathsf{worksAt.Organization})\end{aligned}$$

When $\mathsf{Person}(0)$ [5] is given by evidence, the probability value $P(\mathsf{Professor}(0)) = 0.68$ (we have considered a large number of professors in our experiments), as

[4] http://lattes.cnpq.br.

[5] Indexes $0, 1 \ldots n$ represent individuals from a given domain.

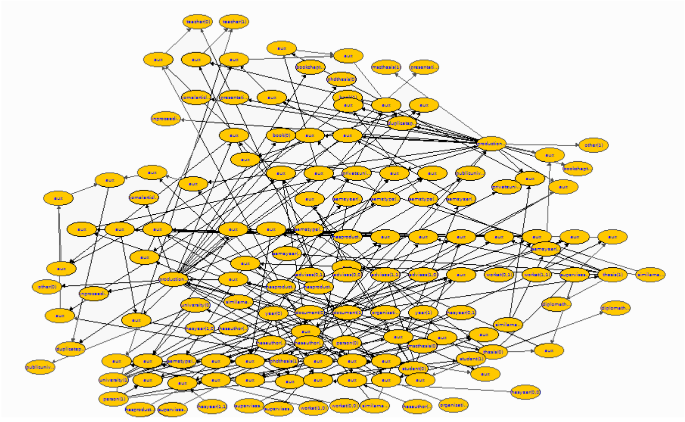


**Fig. 3.** Relational Bayesian network for the Lattes curriculum dataset.

further evidence is given the probability value changes to:

$$P(\mathsf{Professor}(0)|\exists\mathsf{hasPublication}(1)) = 0.72,$$

and

$$P(\mathsf{Professor}(0)|\exists\mathsf{hasPublication}(1) \sqcup \exists\mathsf{advises}(1)) = 0.75.$$

The former concept definition can conflict with standard ILP approaches, where a more suitable definition might be mostly based on conjuntions. In contrast, in this particular setting, the probabilistic logic approach has a nice and flexible behavior. However, it is worth noting that terminological constructs basically rely on the refinement operator used during learning.

Another query, linked to relational classification, allows us to prevent duplicate publications. One can be interested in retrieving the number of publications for a given research group. Whereas this task might seem trivial, difficulties arise mainly due to multi-authored documents. In principle, each co-author would have a different entry for the same publication in the Lattes platform, and it must be emphasized that each entry is be prone to contain errors. In this sense, a probabilistic concept for duplicate publications was learned:

$$\begin{aligned}\mathsf{DuplicatePublication} \equiv\ & \mathsf{Publication}\\ & \sqcap(\exists\mathsf{hasSimilarTitle.Publication} \sqcup \exists\mathsf{hasSameYear.Publication}\\ & \sqcup\mathsf{hasSameType.Publication}))\end{aligned}$$

It clearly states that a duplicate publication is related to publications that share similar title[6], same year and type (journal article, chapter book and so on). At first, the prior probability is low: $P(\mathsf{DuplicatePublication}(0)) = 0.05$. Evidence on title similarity increases considerably the probability value:

$$P(\mathsf{DuplicatePublication}(0)|\exists\mathsf{hasSimilarTitle}(0, 1)) = 0.77.$$

[6] Similarity was carried out by applying a “LIKE” database operator on titles.

Further evidence on type almost guarantees a duplicate concept:

$$P(\mathsf{DuplicatePublication}(0)|\exists\mathsf{hasSimilarName}(1) \sqcap \exists\mathsf{hasSameType}(1)) = 0.99.$$

It must be noted that title similarity does not guarantee a duplicate document. Two documents can share the same title (same author), but nothing prevents them from being published on different means (for instance, a congress paper and an extended journal article). Probabilistic reasoning is valuable to deal with such issues.

## 5 Conclusion

In this paper we have presented algorithms that perform learning of both probabilities and logical constructs from relational data for the recently proposed Probabilistic DL CR$\mathcal{ALC}$. Learning of parameters is tackled by the EM algorithm whereas structure learning is conducted by a combined approach relying on statistical and ILP methods. We approach learning of concepts as a classification task; a Noisy-OR classifier has been accordingly adapted to do so.

Preliminary results have focused on learning a probabilistic terminology from a real-world domain — the Brazilian scientific repository. Probabilistic logic queries have been posed on the induced model; experiments suggest that our methods are suitable for learning ontologies in the Semantic Web.

Our planned future work is to investigate the scalability of our learning methods.

## Acknowledgements

The first author is supported by CAPES. The second author is partially supported by CNPq. The work reported here has received substantial support through FAPESP grant 2008/03995-5.

## References

1. Lisi, F.A., Esposito, F.: Foundations of onto-relational learning. In: ILP '08: Proceedings of the 18th International Conference on Inductive Logic Programming, Berlin, Heidelberg, Springer-Verlag (2008) 158–175
2. Iannone, L., Palmisano, I., Fanizzi, N.: An algorithm based on counterfactuals for concept learning in the semantic web. Applied Intelligence **26**(2) (2007) 139–159
3. Lehmann, J., Hitzler, P.: A refinement operator based learning algorithm for the $\mathcal{ALC}$ description logic. In Blockeel, H., Shavlik, J.W., Tadepalli, P., eds.: ILP '08: Proceedings of the 17th International Conference on Inductive Logic Programming. Volume 4894 of Lecture Notes in Computer Science., Springer (2008) 147–160
4. Muggleton, S.: Inductive Logic Programming. McGraw-Hill, New York (1992)
5. Cozman, F.G., Polastro, R.B.: Loopy propagation in a probabilistic description logic. In: SUM '08: Proceedings of the 2nd International Conference on Scalable Uncertainty Management, Berlin, Heidelberg, Springer-Verlag (2008) 120–133

6. Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning). The MIT Press (2007)
7. de Campos, C.P., Cozman, F.G., Ochoa-Luna, J.E.: Assembling a consistent set of sentences in relational probabilistic logic with stochastic independence. Journal of Applied Logic **7** (2009) 137–154
8. Hailperin, T.: Sentential Probability Logic. Lehigh University Press, Bethlehem, United States (1996)
9. Cozman, F.G., Polastro, R.: Complexity analysis and variational inference for interpretation-based probabilistic description logics. In: Conference on Uncertainty in Artificial Intelligence. (2009)
10. Quinlan, J.R., Mostow, J.: Learning logical definitions from relations. In: Machine Learning. (1990) 239–266
11. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo (1998)
12. Vomlel, J.: Noisy-OR classifier: Research articles. Int. J. Intell. Syst. **21**(3) (2006) 381–398
13. Baader, F., Nutt, W.: Basic description logics. In: Description Logic Handbook. Cambridge University Press (2002) 47–100
14. Fanizzi, N., D'Amato, C., Esposito, F.: DL-FOIL concept learning in description logics. In: ILP '08: Proceedings of the 18th International Conference on Inductive Logic Programming, Berlin, Heidelberg, Springer-Verlag (2008) 107–121
15. Mitchell, T.: Machine Learning. McGraw-Hill, New York (1997)
16. Raedt, L.D., Kersting, K.: Probabilistic inductive logic programming. In: Probabilistic ILP - LNAI 4911. Springer-Verlag Berlin (2008) 1–27
17. Muggleton, S., Raedt, L.D.: Inductive logic programming: Theory and methods. Journal of Logic Programming **19-20** (1994) 629–679
18. Kietz, J.U.: Learnability of description logic programs. In: Inductive Logic Programming, Springer (2002) 117–132
19. Rouveirol, C., Ventos, V.: Towards learning in CARIN-$\mathcal{ALN}$. In: ILP '00: Proceedings of the 10th International Conference on Inductive Logic Programming, London, UK, Springer-Verlag (2000) 191–208
20. Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A.: $\mathcal{AL}$-log: integrating Datalog and description logics. Journal of Intelligent and Cooperative Information Systems **10** (1998) 227–252
21. Rosati, R.: DL+log: Tight integration of description logics and disjunctive datalog. In: KR. (2006) 68–78
22. Cohen, W., Hirsh, H.: Learning the CLASSIC description logic: Theoretical and experimental results. In: (KR94): Principles of Knowledge Representation and Reasoning: Proceedings of the Fourth International Conference, Morgan Kaufmann (1994) 121–133
23. Badea, L., Nienhuys-Cheng, S.H.: A refinement operator for description logics. In: ILP '00: Proceedings of the 10th International Conference on Inductive Logic Programming, London, UK, Springer-Verlag (2000) 40–59
24. Lehmann, J.: Hybrid learning of ontology classes. In: Proceedings of the 5th International Conference on Machine Learning and Data Mining. Volume 4571 of Lecture Notes in Computer Science., Springer (2007) 883–898
25. Inuzuka, N., Kamo, M., Ishii, N., Seki, H., Itoh, H.: Top-down induction of logic programs from incomplete samples. In: ILP '96 : 6th International Workshop. Volume 1314 of LNAI., SV (1997) 265–284
26. Landwehr, N., Kersting, K., DeRaedt, L.: Integrating Naïve Bayes and FOIL. J. Mach. Learn. Res. **8** (2007) 481–507

# Position Papers

# BeliefOWL: An Evidential Representation in OWL Ontology

Amira Essaid[1] and Boutheina Ben Yaghlane[2]

[1] LARODEC Laboratory, Institut Supérieur de Gestion de Tunis
essaid_amira@yahoo.fr
[2] LARODEC Laboratory, Institut des Hautes Etudes Commerciales de Carthage
boutheina.yaghlane@ihec.rnu.tn

**Abstract.** *The OWL is a language for representing ontologies but it is unable to capture the uncertainty about the concepts for a domain. To address the problem of representing uncertainty, we propose in this paper, the theoretical aspects of our tool BeliefOWL which is based on evidential approach. It focuses on translating an ontology into a directed evidential network by applying a set of structural translation rules. Once the network is constructed, belief masses will be assigned to the different nodes in order to propagate uncertainties later.*

## 1 Introduction

Many ontology definition languages have been developed to define ontologies in a formal way. Among them the OWL [3] which is based on crisp logic. This language suffers from its lack to represent real domains containing incomplete knowledge or uncertain information. To overcome this, an extension of the OWL seems to be a convenient solution. Many researches find this extension important and try to propose approaches for handling uncertainty in ontology field. For that purpose, two main mathematical theories have been applied: the probability theory ([2],[7]) and the fuzzy sets theory ([4],[6]).

However not all the problems of uncertainty lend themselves to one of these theories. We can find ourselves faced to situations where we are called to represent the total ignorance or the partial one about information concerning classes. This can be resolved by applying the Dempster-Shafer theory [5]. At this stage, we are interested to use this theory and especially we are encouraged to work with the directed evidential networks [1] which are viewed as effective and appropriate graphical representation for uncertain knowledge. Adding to that, the use of conditional belief functions provides a well representation of the uncertainty in the relationships among the variables of a graph.

In this position paper we present our tool BeliefOWL as an approach for extending an OWL ontology with belief functions as well as the translation of this ontology into an evidential network.

[3] http://www.w3.org/2001/sw/webOnt

## 2 Uncertainty in OWL

The OWL is an expressive language for representing classes and the relations between them for a domain of discourse. However the source of information itself can suffer from giving a sufficient information of a concept. Sometimes we can find ourselves unable to express the exact relation existing between classes because of an incomplete knowledge about the domain of discourse or missed values. Uncertainty extension to the OWL is starting to know a considerable focus during the last years.

To cope with uncertain information in OWL extension, we propose the use of the Dempster-Shafer theory [5]. In fact this theory allows assigning beliefs not only to a single element but to a set of elements. Furthermore, it gives the experts the possibility to represent the total ignorance or the partial one about information concerning the classes of an ontology and the relations that may exist between them. Besides, this theory provides a method for combining several pieces of evidence from different sources to establish a new belief by using Dempster's rule of combination.

One of our goal is to translate an OWL taxonomy into a directed evidential network (DEVN). The DEVN is a model introduced in [1] to represent knowledge under uncertainty by using the belief functions. It is defined as a directed acyclic graph (DAG) where the nodes represent variables and the directed arcs linking nodes describe conditional dependence relations between these variables. These relations are expressed by conditional belief functions for each variable given its parents. Two kinds of belief functions are depicted to represent uncertainty in the DEVN: the prior belief function and the conditional belief function. The former concerns the root node and the latter expresses the belief function of a node given the value taken by its parents.

## 3 Presentation of the BeliefOWL

The figure 1 resumes the different steps followed leading to our tool. In fact the BeliefOWL has as input an OWL ontology and as output a directed evidential network (DEVN).

**Step 1: A Belief Extension to OWL:** An OWL ontology can define classes, properties and individuals. In this paper we will focus on attributing belief masses to the different classes of an OWL taxonomy. For this purpose, we define some new classes able to represent and to introduce this uncertain information.

- **Prior evidence**: We define two classes to express the prior evidence <*beliefDistribution*> and<*priorBelief*>. The former is used to enumerate the different masses related to the different classes of an OWL taxonomy. It has an object property <*hasPriorBelief*> that specifies the relation between classes

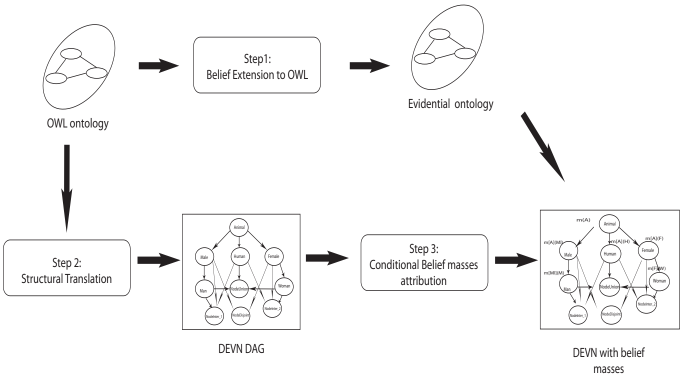


**Fig. 1.** BeliefOWL Framework

<*beliefDistribution*> and <*priorBelief*>. The latter expresses the prior evidence and has a datatype property <*massValue*> which enables to assign a mass value between 0 and 1.

- **Conditional evidence**: It is defined through two main classes <*beliefDistribution*> and<*condBelief*>. The former is the same as in the case of prior evidence but has an object property <*hasCondBelief*>. The latter identifies the conditional evidence and has a datatype property<*massValue*>.

**Step 2: Constructing an Evidential Network**: Given an OWL ontology, we translate it in a DAG by specifying the different nodes to be created as well as the relations existing between these nodes. The construction of the DAG interests some of the OWL statements those related to classes.

- <*owl:class*>: It is represented as a variable node in the translated DEVN.
- <*rdfs:subClassOf*>: When a class is a subclass of another one, a directed arc is drawn from the superclass node to the child subclass node.
- <*owl:disjointWith*>,<*owl:equivalentClass*>:When two classes are related to each other by one of these statements, a new node is created in the translated DEVN and a directed arc is drawn between the two classes and the node added.
- <*owl:intersectionOf*>: A class C may be defined as the intersection of some classes $C_i$(i,. . . ,n). This can be represented in the translated DEVN by an arc from each $C_i$ to C and another one from C and each $C_i$ to a new node created for representing the intersection.
- <*owl:unionOf*>: A class C may be defined as the union of some classes $C_i$(i,. . . ,n). This can be represented in the translated DEVN by an arc from C to each $C_i$ to C and another one from C and each $C_i$ to a new node created for representing the union.

**Step 3: Evidence Attribution**: Once the DAG of our network is constructed, the remaining issue is to assign masses for each node of the network. Considering the DAG that we have got, we can depict two kinds of nodes:

- **ClassesNodes**: are the nodes representing the different classes of our taxonomy and defined by <owl:class>. To this kind of nodes we attribute the prior belief functions and the conditional ones given into the evidential ontology.
- **ConstNodes**: are those related to the constructors of our taxonomy without considering <rdfs:subClassOf> because this kind of constructor is not represented by a specific node. Concerning the constNodes, masses will be attributed according to the constructor we are talking about. In fact if we have a node created to depict an intersection between two classes, the mass will be attributed by applying the Dempster's rule of combination. Concerning the node representing an union, the disjunctive rule of combination will be applied in that case.

Once our evidential network is constructed and the masses are assigned to each node a propagation process can be performed.

## 4 Conclusion

In this paper, we have presented the beliefOWL which is a new approach for representing uncertainty in an OWL ontology. We considered only the case for including uncertainty in classes. This uncertainty is modeled via the Dempster-Shafer theory of evidence. We have presented the theoretical aspects of our tool which consists on translating an OWL ontology into a network. For this purpose, we extend the OWL ontology classes with belief masses, then we apply structural translation rules in order to get a DAG of a directed evidential network. The masses added to the ontology will be extracted and will be attributed to the network's nodes classes.

Further work can carry about the properties and the individuals. The prior beliefs assigned to the different nodes of the network are given by an expert, in the future the assignment can be done automatically through a learning process.

## References

1. Ben Yaghlane,B.: Uncertainty representation and reasoning in directed evidential networks, PhD thesis, Institut Supérieur de Gestion de Tunis Tunisia, 2002.
2. Ding,Z.: BayesOWL: A Probabilistic Framework for Semantic Web, PhD thesis, University of Maryland, Baltimore Country, (2005).
3. Fukushige,Y.: Representing probabilistic relations in RDF, proc. of Work. on URSW at the 4th ISWC, Galway, Ireland, (2005).
4. Gao,M., Liu,C.: Extending OWL by fuzzy description logic, proc. of the 17th IEEE ICTAI05, Hong Kong, China, 562-567 (2005).
5. Shafer,G.: A Mathematical Theory of Evidence, Princeton University Press (1976).
6. Stoilos,G., Stamou,G., Tzouvaras,V., Pan,J., Horrocks,I.: Fuzzy OWL: Uncertainty and the Semantic Web, proc. of the Inter. Work. on OWL-ED05, Galway, Ireland, (2005).
7. Yang,Y., Calmet,J.: OntoBayes: An Ontology-Driven Uncertainty Model, proc. of CIMSCA/IAWTIC, Washington, DC, USA. IEEE Computer Society, 457-463 (2005).

# Fuzzy Taxonomies for Creative Knowledge Discovery

Trevor Martin[1,2], Zheng Siyao[1,3] and Andrei Majidian[2]

[1] AI Group, University of Bristol, BS8 1TR UK
[2] Intelligent Systems Lab, BT, Adastral Park, Ipswich IP5 3RE, UK
[3] School of Computer Science and Engineering, BeiHang University, Beijing, China

Trevor. Martin@bristol.ac.uk, zhengsyao@gmail.com, Andrei .Majidian@bt.com

**Abstract.** A systematic form of creative knowledge discovery is outlined, requiring taxonomies to generalise knowledge structures and mappings between taxonomies to find parallels between knowledge structures from different domains. These share many of the features needed to handle uncertainty in the semantic web, and results will be relevant to the URSW community.

**Keywords:** fuzzy taxonomy, creative knowledge discovery, fuzzy association rules, uncertainty in semantic web

## 1 Introduction

Almost by definition, creative knowledge discovery is difficult to automate and harder to assess objectively. By creative knowledge discovery, we mean finding previously unknown links between concepts or small "chunks" of knowledge in such a way that useful additional knowledge is generated. It can be distinguished from "standard" knowledge discovery by defining the latter as the search for explanatory and/or predictive patterns and rules in large volume data within a specific domain. For example, a knowledge discovery process might examine an ISP(internet service provider)'s customer database and determine that people who have a high monthly spend and who send more than three emails to the support centre in a single month are very likely to change to a different provider in the following month. Such knowledge is implicit within the data but is useful in predicting and understanding behaviour.

By contrast, creative knowledge discovery is more concerned with "thinking the unthought-of" and looking for new links, new perspectives, etc. Such links are often found by drawing parallels between different domains and looking to see how well those parallels hold - for example, compare the ISP example mentioned above to a hotel chain finding that regular guests who report dissatisfaction with two or more stays often cease to be regular guests unless they are tempted back by special treatment (such as complimentary room upgrades). This is a simple illustration of similar problems (losing customers) in different domains. A solution in one domain (complimentary upgrades) could inspire a solution in the second (e.g. a higher download allowance at the same price). Of course, such analogies may break down when probed too far but they often provide the creative insight necessary to spark a new solution through a new way of looking at a problem. In many cases, this inspiration is often referred to as “serendipity”, or accidental discovery.

It is possible that many serendipitous discoveries are subsequently rationalised as the outcome of rigorous application of the scientific process. The traditional view of the scientist is as a generator and tester of hypotheses - often this is presented as an almost mechanical process and systems such as King’s robot scientist [1] take this to an extreme, using an inductive logic programming approach to systematically generate and test hypotheses in a laboratory.

In this paper we outline a project to automate creative knowledge discovery. The aim is to find parallels between different knowledge repositories - in this case, semantically annotated networks of documents or process models - in the hope of transferring useful links from one network to another. In the case of process models from different domains, the aim is to identify possible improvements in one process if its analogue in the other domain is more efficient in some way.

This work shares many of the problems faced by research into uncertainty in the semantic web - the mapping between repositories is very similar to a mapping between ontologies, and the creation of knowledge networks encounters several issues that are well-known from the semantic web, such as the need for imprecise concepts, integration of sources that represent entities and classes at different levels of detail etc. The work is at an early stage, and this paper briefly outlines (i) a possible approach to automating creativity which relies on the use of fuzzy taxonomies and (ii) preliminary work on automatic extraction of taxonomies from data; this requires a representation of uncertainty similar to that needed for the semantic web.

## 2 A Method for Creative Knowledge Discovery

Can creativity - in this sense of suddenly making novel connections - be automated? Koestler [2] summarised this view of creativity as follows:

*“The creative act is not an act of creation in the sense of the Old Testament. It does not create something out of nothing: it uncovers, selects, re-shuffles, combines, synthesizes already existing facts, idea, faculties, skills. The more familiar the parts, the more striking the new whole”*

Table 1 - attributes of two music players (taken from [4])

| Conventional tape recorder | Sony Walkman |
|---|---|
| big | small |
| clumsy | neat |
| records | does not record |
| plays back | plays back |
| uses magnetic tape | uses magnetic tape |
| tape is on reels | tape is in cassette |
| speakers in cabinet | speakers in headphones |
| mains electricity | battery |

Sherwood [3] proposes a systematic approach, in which a situation or artefact is represented as an object with multiple attributes, and the consequences of changing attributes, removing constraints, etc are progressively explored. For example, given an old style reel-to-reel tape recorder as starting point, Sherwood's approach is to list some of its essential attributes, substitute plausible alternatives for a number of these attributes, and evaluate the resulting conceptual design or solution. Table 1 shows how this could have led to the Sony Walkman in the late 70s [4]. Again, with the benefit of hindsight the reader should be able to see that by changing magnetic tape to a hard disk and considering the way music is purchased and distributed, the same method could (retrospectively, at least) lead one to invent the iPod. Of course, having the vision to choose new attributes and the knowledge and foresight to evaluate the result is the hard part - and the creative steps are usually only obvious with hindsight.

This systematic approach is ideally suited to handling data which is held in an object-attribute-value format, provided we have a means of changing/generalising attribute values. We intend to use taxonomies for this purpose, so that "sensible" changes can be made (e.g. *mains*, *battery* are both possible values for a *power* attribute). Representing an object O as a set of attribute-value pairs $\left\{(a_i, v_i) \middle| attribute\ a_i\ of\ object\ O\ has\ value\ v_i\right\}$ we generate a new "design" $O^* = \left\{(a_i, T(v_i))\right\}$ by changing one or more values using $T_i$, a non-deterministic transformation of a value to another value from the same taxonomy. Given sufficient time, this would simply enumerate all possible combinations of attribute values. We can reduce the search space by looking at the solution to an analogous problem in a different domain.

Our aim is to adapt previously developed tools for taxonomy matching [5] so that analogies can be found; the next section briefly outlines a way to extract taxonomic structure when it is not explicitly available.

## 3 Extracting Embedded Soft Taxonomies

An ontology essentially consists of a taxonomy of concepts, one or more relations between concepts, and rules which impose constraints and allow data transformation. The idea of an ontology is central to the semantic web [6], although there can be a very high cost in creation and maintenance. This is reflected in practical experience - it is rare to find web-based data that is fully marked up with RDF or OWL metadata. It is far more common to encounter data that is stored in a relational database or an equivalent XML-tagged format. Such data often contains implicit taxonomies - a relational table may flatten hierarchical data into one or more attributes. For example, a film database may record genre(s) and sub-genre(s) as separate fields, hiding the hierarchical dependency. The hierarchy may be obvious to a human reader of the data, but it is invisible to the machine. Similarly, XML tags can hide structure. XML relies on human interpretation for its "semantics" - a programmer can take advantage of the fact that <*iPod*> and <*walkman*> are subtypes of <*music Player*>, but a program has no way of knowing this unless it is made explicit by means of a taxonomy. Although a well-designed schema will make hierarchical structure explicit, our experience is that a significant proportion of data sources rely on programmer intuition instead.

We have investigated formal concept analysis (FCA) [7, 8] as a way of extracting hidden structure from a dataset in object-attribute-value form. In its simplest form, FCA considers a binary-valued table, where each row corresponds to an object and each column to an attribute (property). The extension to a fuzzy case is (relatively) straightforward, by considering a fuzzy relation *R** and alpha-cuts which reduce the problem to the crisp case. A brief outline and promising initial results are given in [9].

## 4 Applications

Two specific domains form demonstrators for this work. XML process mining algorithms exist to discover process model from log files; various additions include heuristic and fuzzy approaches to handle noisy data. Semantic processing mining involves ontology knowledge. The ProM [www.processmining.org] platform takes SA-MXML (semantic annotated mxml) files as input, where the annotation conforms to the Web Service Modelling Language. The aim of this demonstrator is to find (partial) similarities between process models in different domains, and use process simulation tools to determine whether one process can be improved by slightly altering it to match the second process more closely. The second demonstrator is based on web forum discussions and support centre documentation, and will attempt to improve the automated provision of "help" information.

## 5 Summary

This paper has briefly outlined a project to automate aspects of creative knowledge discovery. The project is in early stages. Although not a direct application of uncertain reasoning in the semantic web, it shares many of the same problems and useful cross-fertilisation of ideas should be possible.

**Acknowledgement** : this work was partly funded by the FP7 BISON (Bisociation Networks for Creative Information Discovery) project, number 211898.

## 6 References

1.King, R.D., et al., *Functional genomic hypothesis generation and experimentation by a robot scientist*. Nature, 2004(6971): p. 247-251.

2.Koestler, A., *The Act of Creation*. 1964: Macmillan. 751.

3.Sherwood,D. *SilverBullet Machine::Guide to Creativity*,2009, www.silverbulletmachine.com

4.Sherwood, D., *Koestler's Law: The Act of Discovering Creativity-And How to Apply It in Your Law Practice*. Law Practice, 2006. **32**(8).

5.Martin,T.P, B.Azvine, Y.Shen, *Granular Assoc Rules for Multiple Taxonomies: A Mass Assignment Approach* in *Uncertain Reasoning in the Sem Web*, M.Nickles, Ed 2008,Springer.

6.Berners-Lee, T, J.Hendler, and O.Lassila, *Semantic Web*, in *Scientific American* 2001 p28-37.

7.Ganter, B, R. Wille, *Formal Concept Analysis: Mathematical Foundations*. 1998: Springer.

8.Priss, U., *Formal Concept Analysis in Information Science*.

9.Majidian, A. and T.P. Martin. *Extracting Taxonomies from Data - a Case Study using Fuzzy FCA*. in *Web Intelligence-09*. 2009. Milan, Italy: IEEE Computing.

# Position paper: Uncertainty reasoning for linked data

Dave Reynolds[1]

[1] Hewlett Packard Laboratories, Bristol
Dave.e.Reynolds@gmail.com

**Abstract.** Linked open data offers a set of design patterns and conventions for sharing data across the semantic web. In this position paper we enumerate some key uncertainty representation issues which apply to linked data and suggest approaches to tackling them. We suggest the need for vocabularies to enable representation of link certainty, to handle ambiguous or imprecise values and to express sets of assumptions based on named graph combinators.

**Keywords:** Uncertainty reasoning; linked open data; semantic web

## 1 Introduction

The need for reasoning over uncertain information within the semantic web occurs in many different situations. It can arise from intrinsic uncertainty in the world being modeled or from limitations of the sensing or reasoning agent itself (epistemic). The term *uncertainty* is often used to refer to many different notions including ambiguity, randomness, vagueness, inconsistency, incompleteness [1][9].

In recent years an approach to the semantic web, called *linked data*, has been developed and offers a promising route to practical and widespread semantic web uptake. It provides a set of design guidelines or patterns for how the semantic web technologies, and broader web architecture, can be used for sharing information. The existing guidelines and practices have no provision for representation of uncertainty; yet linked data is indeed fraught with many of these different types of uncertainty.

In this brief position paper we examine the ways in which uncertainty can occur in a linked data setting and sketch possible approaches to addressing the issues raised.

## 2 Linked data

*Linked Data* is a set of conventions for publishing data on the semantic web. It is based on principles outlined by Tim Berners-Lee [2]. These principle advocate the use of http URIs for naming entities, the publication of data about these URIs using the standards (RDF, SPARQL) and inclusion of  links to other URIs so that agents can discover more information. While quite simple these guidelines, along with a growing body of practical advice [3], have led to publication and linking of many datasets in this form [4]. This has resulted in high profile commercial applications such as [5].

While not explicitly stated, the style of linked data places an emphasis on data sharing and simplicity, with corresponding less emphasis on depth of modeling and reasoning. Yet the intrinsic nature of the linked data approach leads to issues of uncertainty representation and reasoning. This is due to the emphasis on cross-linking multiple data sources that have been independently developed and modeled. Uncertainty can arise from the instance linking process, from the mapping between different sources models and due to differing hidden assumptions in the underlying datasets. Yet the essence of linked data, and a large part of the reason for its uptake, is simplicity. The data is intended to be self-descriptive and accessible through simple link following and graph union or through SPARQL endpoints. Our challenge is to develop a common, easy to deploy, approach to uncertainty representation which can be applied to linked data sets without losing this simplicity.

## 3 Some sources of uncertainty in linked data applications

In this section we enumerate some key sources of uncertainty for linked data. We focus on the sources which directly result from the intrinsic nature of linked data – the cross-linking of independently developed RDF datasets.

### 3.1 Ambiguity resulting from data merging

In linked data, entities (Individuals) which co-occur with different URIs in different datasets are unified. This is achieved by publishing `owl:sameAs` relations between identified entities, either within the dataset or as a separate link set. The process of identifying such co-references is imperfect. Firstly, the co-references are typically found by a mixture of string matching, attribute matching, and type constraints, generally based on a statistical or machine learning algorithm [6]. Thus co-references are only identified with some probability (or less formal heuristic weighting). Yet the asserted links are binary and the strength of association is lost. Secondly, the nature of the entities is ambiguous in some datasets. For example, Wikipedia and thus DBPedia conflate the concepts of the City *Bristol* in the UK and the associated Unitary Authority. A co-reference link that identified the ambiguous DBPedia concept with one that specifically denotes the Unitary Authority would be an error in general, even though it may be an acceptable approximation in some situations.

### 3.2 Misalignment of precision and assumptions between merged sources

Many datasets in the linked data web publish property values for the entities they describe; for example, the *population* of the *City of London*. Yet those values are sometimes imprecise or dependent upon measurement assumptions that are not made explicit. For example, the *population* of a city depends on the time of the measurement, the measurement methodology and the precise definition of the boundary of the city; it is also subject to statistical uncertainty. As a result, at the time of writing, a linked data query on London returns a graph with four assertions on its population ranging from 7,700,000 to 8,500,000. One of these sources of variation, the time of measurement, is sometimes made explicit in data and indeed one of the four assertions is (indirectly) time qualified. However, such contextual qualification is not consistently available and, in any case, only accounts for one source of variation. Thus when datasets are linked the resulting union will often have multiple conflicting values for supposedly functional properties.

### 3.3 Misalignment of models

When linking datasets we also want to map the associated ontologies. This process is just as error prone as entity co-reference since the axiomatization of concepts in the ontologies is rarely so complete as to allow a unambiguous mapping. Errors in the ontology mapping can lead to global effects such as unexpected identification of related concepts. Determining and publishing such alignment errors is the subject of considerable research and is outside the scope of this paper.

### 3.4 Absence of source reliability information

Separate from the uncertainty arising from combination and linking of datasets then the datasets themselves can be uncertain or contain errors (either accidental or malicious). While this is true in general in the semantic web, the linked data approach implies broad cross linking with no provision for narrow scoping of link references. This exacerbates the problems of the veracity or trustworthiness of included datasets.

## 4 Mitigation approaches

We now discuss approaches to mitigate the effects of these uncertainty sources on the consumers of linked data. In keeping with linked data methodology we seek simple, broadly applicable, design patterns. In particular, we suggest the need for design patterns for making the uncertainty inherent in the linked datasets more explicit, and mechanisms to enable selective combination of datasets (so that problematic values or links can be omitted). In this a short position paper we only sketch the suggested approaches as a basis for discussion in the workshop.

### 4.1 Link vocabulary

The *link vocabulary* would provide a common representation for co-reference links, enabling publication of the link certainty information on which per-link inclusion decisions can be made. This can be achieved by extending the voiD ontology [8] with a concept *UncertainLinkSet* (as a subclass of `void:LinkSet`), and associated properties for describing the method used for deriving the link set. The *UncertainLinkSet* itself would contain n-ary relations (*WeightedLink*) comprising the link and associated link weight. Different subclasses of *WeightedLink* indicate different interpretations of the link weight (such as probabilistic or ad hoc).

### 4.2 Imprecise value vocabulary

The *imprecise value vocabulary* would provide a common representation for imprecise values that arise from data set merger, as discussed in 3.2. This would allow republication of merged datasets which explicitly show the variation in source data values. Returning to our example of the population of London the merged set might look like:

```
:London :population [a :ImpreciseValue;
   :samplevalue [:value 7700000; :source :s1; :context :y2009]
   :samplevalue [:value 7900000; :source :s2; :context :y2008]
   :estimatedValue 785123]
```

### 4.3 Override graphs

Finally we suggest the need for override graphs so that one agent can publish retractions and overrides to the link assertions or data assertions made by another.

The current approach to this, in linked data applications, is to partition data and link sets into named graphs [7]. For example, rather than include all the co-reference links directly in the same graph as the entity descriptions, we partition them into a separate named graph. In this way a RESTful access can see the union of the relevant graphs but a SPARQL endpoint can support selection of which graphs to include. This allows agents to avoid selected link sets or sub-sources but only at the grain size of the entire graph. To overcome this limitation we suggest extending the VoiD vocabulary to include graph combinators *difference, union* and *replace*. So one source can decide which subsets of the data and links to trust, and can then publish the assumptions it is making as a set of deltas over the source graphs. The difference graphs enable per-link and per-assertion changes to be expressed even if the underlying source only publishes the link set or data assertions as monolithic graphs.

## 5 Discussion

Of the issues in section 3 we have suggested an agenda for how to address some of them. The *link* and *imprecise value* vocabularies enable publication of link uncertainty (3.1) and value ambiguity (3.2) information in linked data sets. The vocabularies themselves would not remove the uncertainties, nor the problems of estimating them. However, simply having a means to publish this information is already a step forward. The suggested *graph combinators* would enable an agent to make and publish more selective data combinations, based on its interpretation of link strengths and data values. This does not solve the problems of deciding which parts of which sources to trust, but it does enable more effective sharing of such decisions.

## References

1. Laskey, K.J., et al.: Uncertainty Reasoning for the World Wide Web, W3C Incubator Group Report, 31 March 2008. http://www.w3.org/2005/Incubator/urw3/XGR-urw3/
2. Berners-Lee, T.: Linked Data. http://www.w3.org/DesignIssues/LinkedData.html (2006)
3. http://linkeddata.org/
4. http://linkeddata.org/data-sets
5. Kobilarov, G., et al.: Media Meets Semantic Web --- How the BBC Uses DBpedia and Linked Data to Make Connections. Proc. of the 6th European Semantic Web Conference on the Semantic Web: Research and Applications (Heraklion, Crete, Greece). Lecture Notes In Computer Science, vol. 5554. Springer-Verlag, Berlin, Heidelberg, 723-737. (2009)
6. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate Record Detection: A Survey. IEEE Transactions on Knowledge and Data Engineering 19 (2007)
7. Carroll, J. J., Bizer, C., Hayes, P., and Stickler, P.: Named graphs, provenance and trust. In Proceedings of the 14th international Conference on World Wide Web (Chiba, Japan, May 10 - 14, 2005). WWW '05. ACM, New York, NY, 613-622. (2005)
8. Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: voidD Guide : Using the vocabulary of Interlinked Datasets. http://rdfs.org/ns/void-guide (2009)
9. Kruse, R., Schwecke, E., and Heinsohn, J. 1991 *Uncertainty and Vagueness in Knowledge Based Systems*. Springer-Verlag New York, Inc.

# Author Index