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.

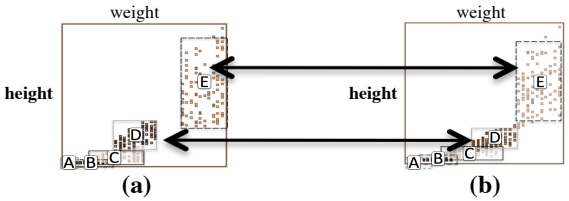

Fig 1. Classifying instances using concepts of different ontologies based on a pair of attributes *weight* and *height*, reveal similarity correlation between the same pair of attributes, in separate ontologies (a) and (b).

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 $\mathcal{SHOQ}$(**D**) extension has added the capability to specify qualified number restrictions, and the $\mathcal{SHOIN}$(**D**) extension has combined singleton classes, inverse roles and unqualified number restrictions. Further, $\mathcal{SHOIN}$(**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-$\mathcal{SHOQ}$ (**D**)[15], a probabilistic extension to $\mathcal{SHOQ}$ (**D**) [18][24], fuzzy $\mathcal{SHOIN}$ (**D**) [26], a fuzzy extension to $\mathcal{SHOIN}$ (**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-$\mathcal{SHOQ}$ (**D**) has provided a sound, complete, and decidable reasoning technique for probabilistic Description Logics. Fuzzy $\mathcal{SHOIN}$ (**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 $\mathcal{SHOIN}$ (**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 $\mathcal{SHOIN}$ (**D**) is introduced as an extension to fuzzy $\mathcal{SHOIN}$ (**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 $\geq$ 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].

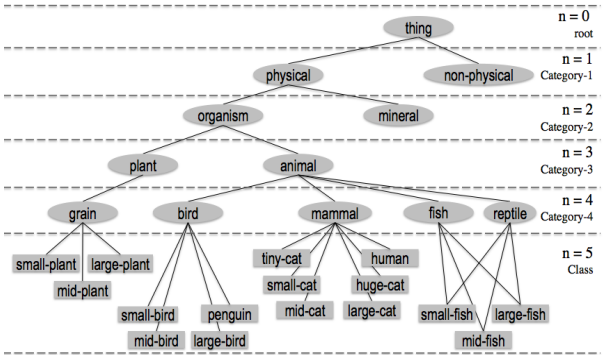

Fig 2. An ontology, split by levels *n*, which are used for iterating edges in our algorithm in section 3.2.

[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; $i = \{ 0, \ldots, f \}$

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, ... , $levels(\mathcal{O}_h)$ }; *where* $n = 0$ is the tree root
c_{n_j} := concept $\in \mathcal{O}$ at level n; where $j = \{ 0, ... , |C_n| \}$
$|c|$:= 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, $A = \{ a_0, a_1, \ldots, a_f \}$, 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

σ = Bayesian probability of classifying c_{n_j} correctly with a *Gr*.
φ = coverage (number of instances in a *Gr* classifying c_{n_j}) out of $|c|$.
Pr = σ (φ / $|c|$) : 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

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

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

$$\Omega \quad = \text{Probability of } c_{n_j}, \text{ calculated as sum of } c_n \text{ probabilities } (Pr) \text{ with an associated coverage } |c|\ .$$
$$\Omega_j\, Sig_j(c_{n_j}) \quad \leftarrow (Pr_x Gr_x) \vee (Pr_y Gr_y) \vee \ldots \vee (Pr_z Gr_z)\ .$$

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 Ω_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.

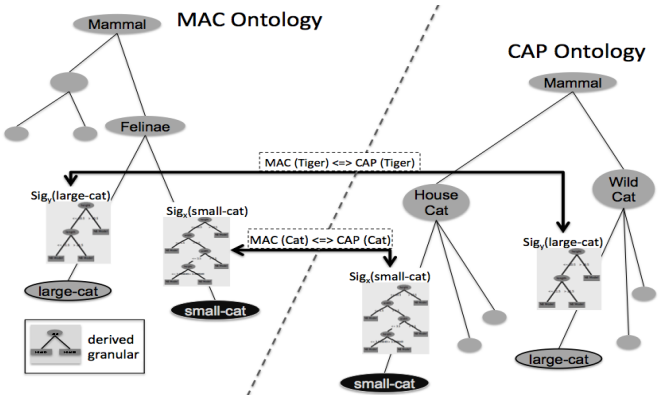

Fig 3. Concepts are mapped using derived *signatures* between two ontologies from section 4.1.1.

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 $F = \{tiny\text{-}cat, small\text{-}cat, mid\text{-}cat, large\text{-}cat, huge\text{-}cat\}$, and derive the DT in Figure 4. Each leaf node represents a Bayesian model for each concept, with various degrees of probability σ and coverage φ, 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

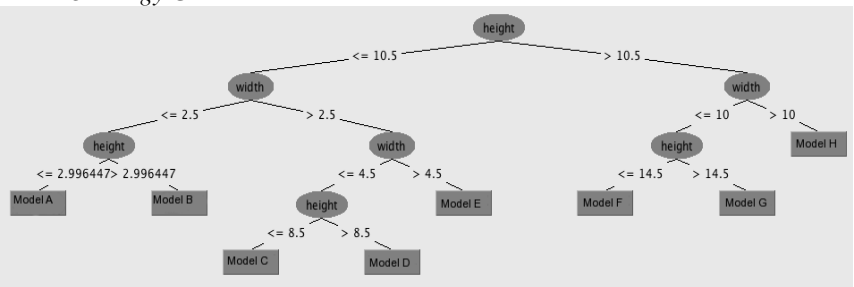

Fig 4. NBTree classifying MAC *Felinae* based on *height*, *width*, weight (omitted) and *country* (omitted).

[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	
	σ	φ		
A	0.89	101	Gr_0(tiny-cat)	← (x ≤ 10.5) ∧ (y ≤ 2.5) ∧ (x ≤ 2.99)
	0.09	9	Gr_1(small-cat)	← (x ≤ 10.5) ∧ (y ≤ 2.5) ∧ (x ≤ 2.99)
B	0.92	44	Gr_2(small-cat)	← (x ≤ 10.5) ∧ (y > 2.5) ∧ (x > 2.99)
C	0.90	34	Gr_3(small-cat)	← (x ≤ 10.5) ∧ (y > 2.5) ∧ (y ≤ 4.5) ∧ (x ≤ 8.5)
D	0.58	13	Gr_4(small-cat)	← (x ≤ 10.5) ∧ (y > 2.5) ∧ (y ≤ 4.5) ∧ (x > 8.5)
	0.29	6	Gr_5(mid-cat)	← (x ≤ 10.5) ∧ (y > 2.5) ∧ (y ≤ 4.5) ∧ (x > 8.5)
E	0.64	6	Gr_6(mid-cat)	← (x ≤ 10.5) ∧ (y > 2.5) ∧ (y > 4.5)
F	0.87	26	Gr_7(mid-cat)	← (x > 10.5) ∧ (y ≤ 10) ∧ (y ≤ 14.5)
G	0.78	93	Gr_8(large-cat)	← (x > 10.5) ∧ (y ≤ 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.

Ω		Signature	
ΣPr	|c|		
0.89	*101*	Sig_0(tiny-cat)	← $(Pr_0 Gr_0)$
0.78	100	Sig_1(small-cat)	← $(Pr_1 Gr_1) \vee (Pr_2 Gr_2) \vee (Pr_3 Gr_3) \vee (Pr_4 Gr_4)$
0.78	*60*	Sig_2(mid-cat)	← $(Pr_5 Gr_5) \vee (Pr_6 Gr_6) \vee (Pr_7 Gr_7) \vee (Pr_9 Gr_9)$
0.78	*93*	Sig_3(large-cat)	← $(Pr_8 Gr_8)$
0.96	*105*	Sig_4(huge-cat)	← $(Pr_{10} Gr_{10})$

4.1.2 CAT Felinae Ontology Granulation

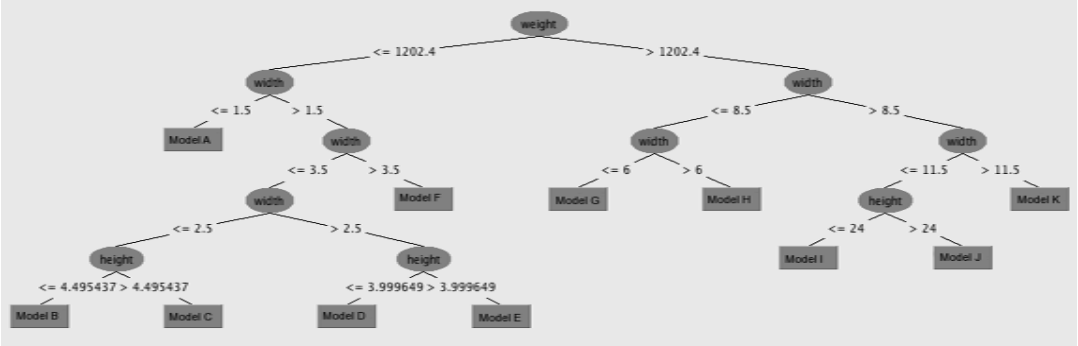

Fig 5. NBTree classifying CAP *Felinae* based on *height*, *width*, *weight*, (omitted) and *country* (omitted).

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

Model	*Pr*		Granule	
	σ	φ		
A	0.51	24	Gr_0(small-cat)	← (weight ≤ 1202.4) ∧ (y ≤ 1.5)
	0.43	20	Gr_1(mid-cat)	← (z ≤ 1202.4) ∧ (y ≤ 1.5)
B	0.09	4	Gr_2(small-cat)	← (z ≤ 1202.4) ∧ (y > 1.5) ∧ (y ≤ 3.5) ∧ (y ≤ 2.5) ∧ (x ≤ 4.5)
	0.85	45	Gr_3(tiny-cat)	← (z ≤ 1202.4) ∧ (y > 1.5) ∧ (y ≤ 3.5) ∧ (y ≤ 2.5) ∧ (x ≤ 4.5)
C	0.38	13	Gr_4(small-cat)	← (z ≤ 1202.4) ∧ (y > 1.5) ∧ (y ≤ 3.5) ∧ (y ≤ 2.5) ∧ (x > 4.5)
	0.54	19	Gr_5(mid-cat)	← (z ≤ 1202.4) ∧ (y > 1.5) ∧ (y ≤ 3.5) ∧ (y ≤ 2.5) ∧ (x > 4.5)
D	0.15	10	Gr_6(small-cat)	← (z ≤ 1202.4) ∧ (y > 1.5) ∧ (y ≤ 3.5) ∧ (y > 2.5) ∧ (x ≤ 4)
	0.80	56	Gr_7(tiny-cat)	← (z ≤ 1202.4) ∧ (y > 1.5) ∧ (y ≤ 3.5) ∧ (y > 2.5) ∧ (x ≤ 4)
E	0.40	15	Gr_8(small-cat)	← (z ≤ 1202.4) ∧ (y > 1.5) ∧ (y ≤ 3.5) ∧ (y > 2.5) ∧ (x > 4)
	0.53	20	Gr_9(mid-cat)	← (z ≤ 1202.4) ∧ (y > 1.5) ∧ (y ≤ 3.5) ∧ (y > 2.5) ∧ (x > 4)

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

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

Ω		Signature	
ΣPr	|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-cat) ← (width > 0) ∧ (width ≤ 4) ∧ (height > 4) ∧ (height ≤ 8) ∧ (weight > 20) ∧ (weight ≤ 50)*, could represent such clusters. A definition could be built by classifying a *Class* with a single attribute, like *Sig(mid-cat) ← ((width > 0)∧(width ≤ 0.7)) ∨ ((width > 1.1)∧(width ≤ 2.1)) ∨ ((width > 3.4)∧(width ≤ 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 (*tiny-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).

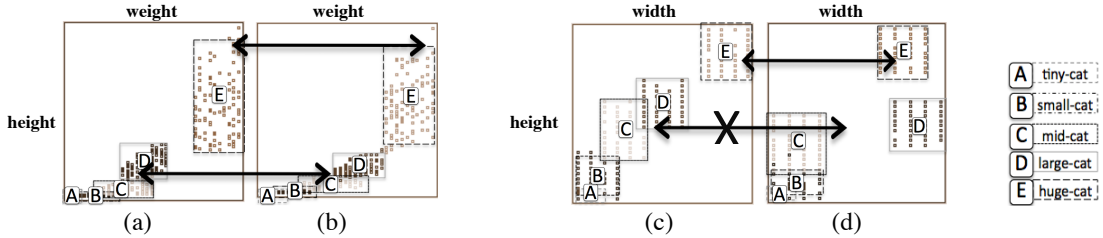

Fig 6. Attribute associations: *weight* × *height* for (a) MAC, (b) CAP; *height* × *width* for (c) MAC, (d) CAP.

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)