Towards Evidence-Based Terminological Decision Trees

Giuseppe Rizzo, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito

LACAM – Università degli studi di Bari "Aldo Moro", Bari, Italy
name.lastname@uniba.it

Abstract. We propose a method that combines terminological decision trees and the Dempster-Shafer Theory, to support tasks like *ontology completion*. The goal is to build a predictive model that can cope with the epistemological uncertainty due to the Open World Assumption when reasoning with Web ontologies. With such models not only one can predict new (non derivable) assertions for completing the ontology but by assessing the quality of the induced axioms.

1 Introduction

In the context of machine learning applied to Web ontologies, various methods have been proposed in order to predict new assertions. It has turned out that the models resulting from these methods can often provide effective reasoning services which are comparable to those offered by reasoners [1].

Focusing on the *instance-checking* problem [2], i.e. the task of assessing the class-membership for individual resources, it is well known that a reasoner may be unable to prove the membership of an individual to a given concept (or to its complement). This is often caused by flaws introduced in the ontology construction phase owing to lacking disjointness axioms. The same problem may appear also with logic-based predictive classification models produced by machine learning algorithms, such as the *Terminological Decision Trees* [3] (TDTs), a specialization of first-order decision trees [4]. In this work we extend the scope of TDTs by employing the *Dempster-Shafer Theory* [5] (DST) because, differently from the instance-based approach proposed in [6], logic-based classification models generally do not provide an epistemic uncertainty measure. This may be very important as a quality measure for predicted assertions in related problems such as data integration in the context of Linked Data[1], where it could contribute as a measure of *provenance* [7]. Purely logical models cannot handle properly cases of tests resulting in an unknown membership. The uncertainty is not explicitly considered when an individual is classified w.r.t. a given test class. The situation is similar to the case of missing values in prediction with (propositional) decision trees. The underlying idea of the proposed extension is to exploit standard algorithms to cope with missing values for a test by partitioning the observation w.r.t. all possible values of the test and then following all branches. Once the leaves are reached, the results are combined. Thanks to the combination rules used to pool evidences [5], the DST is a more suitable framework than the Bayesian theory of probability to cope with epistemic uncertainty

[1] http://linkeddata.org

A. Laurent et al. (Eds.): IPMU 2014, Part I, CCIS 442, pp. 36–45, 2014.

and ignorance related to the Open World Assumption (OWA) that characterizes Web ontologies.

The DST has been integrated in various algorithms [8,9] with results that are competitive with the classical version. So, we want to investigate if this model can be used also in machine learning algorithm for the Semantic Web in order to obtain better results of classifiers in terms of predicted assertions.

The paper is organized as follows: Section 2 introduces basics concerning the concept learning task in *Description Logic knowledge bases* and describes the original version of TDTs; in Section 3 the algorithm for inducing a TDT based on the DST is proposed while in Section 4 an early-stage empirical evaluation is described; finally, further extensions of this work are described.

2 Background

Knowledge Bases. In Description Logics (DLs) [2], a domain is modeled through primitive *concepts* (classes) and *roles* (relations), which can be used to build complex descriptions regarding *individuals* (instances, objects), by using specific operators that depend on the adopted language. A *knowledge base* is a couple $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ where the *TBox* $\mathcal{T}$ contains axioms concerning concepts and roles (typically inclusion axioms such as $C \sqsubseteq D$) and the *ABox* $\mathcal{A}$ contains assertions, i.e. axioms regarding the individuals ($C(a)$, resp. $R(a, b)$). The set of individuals occurring in $\mathcal{A}$ is denoted by $\mathsf{Ind}(\mathcal{A})$.

The semantics of concepts/roles/individuals is defined through interpretations. An *interpretation* is a couple $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is the *domain* of the interpretation and $\cdot^{\mathcal{I}}$ is a *mapping* such that, for each individual a, $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, for each concept C, $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and for each role R, $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The semantics of complex descriptions descends from the interpretation of the primitive concepts/roles and of the operators employed, depending on the adopted language. $\mathcal{I}$ satisfies an axiom $C \sqsubseteq D$ (C is subsumed by D) when $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ and an assertion $C(a)$ (resp. $R(a, b)$) when $a^{\mathcal{I}} \in C^{\mathcal{I}}$ (resp. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$). $\mathcal{I}$ is a *model* for $\mathcal{K}$ iff it satisfies each axiom/assertion α in $\mathcal{K}$, denoted with $\mathcal{I} \models \alpha$. When α is satisfied w.r.t. these models, we write $\mathcal{K} \models \alpha$.

We will be interested in the *instance-checking* inference service: given an individual a and a concept description C determine if $\mathcal{K} \models C(a)$. Due to the *Open World Assumption* (OWA), answering to a class-membership query is more difficult w.r.t. *Inductive Logic Programming* (ILP) settings where the closed-world reasoning is the standard. Indeed, one may not be able to prove the truth of either $\mathcal{K} \models C(a)$ or $\mathcal{K} \models \neg C(a)$, as there may be possible to find different interpretations that satisfy either cases.

Learning Concepts in DL. The concept learning task in DL can be defined as follows. Given:

- a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$
- a target concept C,
- the sets of positive and negative examples for C:
  $Ps = \{a \in \mathsf{Ind}(\mathcal{A}) \mid \mathcal{K} \models C(a)\}$ and $Ns = \{a \in \mathsf{Ind}(\mathcal{A}) \mid \mathcal{K} \models \neg C(a)\}$

the goal is to obtain a concept description D for C ($C \sqsubseteq D$), such that:

- $\mathcal{K} \models D(a) \quad \forall a \in Ps$
- $\mathcal{K} \models \neg D(a) \quad \forall a \in Ns$

In order to investigate about the learning of multiple disjoint concepts, the formulation of the problem is more restrictive than the one proposed in [3], where the negative examples were the individuals a for which $\mathcal{K} \not\models D(a)$. The resulting concept descriptions can be used to solve an *instance-checking* problem for new individuals. Similarly to a *First Order Logic Decision Tree*, a binary tree where each node contains a conjunction of literals and each variable that is introduced in a node cannot appear in the right branch of that node, a Terminological Decision Tree can be defined as follows:

Definition 1 (Terminological Decision Tree). *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a* Terminological Decision Tree *is a binary tree where:*

- *each node contains a conjunctive concept description D;*
- *each departing edge is the result of a class-membership test w.r.t. D, i.e., given an individual a, $\mathcal{K} \models D(a)$?*
- *if a node with E is the father of the node with D then D is obtained by using a refinement operator and one of the following conditions should be verified:*
  - *D introduces a new concept name,*
  - *D is an existential restriction,*
  - *D is an universal restriction of any its ancestor.*

However, a set of concept descriptions is generated by means of the refinement operator and the best one is chosen to be installed as a child node. The best description is the one that maximizes a purity measure respect to the previous level [3]. The measure may be defined as *accuracy* in a binary classification problem, *purity* $= p/(p+n)$, where p is the number of positive examples and n the number of negative ones reaching a node.

3 Induction of the Terminological Trees

The method for inducing TDTs based on the DST uses a divide-and-conquer strategy. It requires the target concept C, a training set $Ps \cup Ns \cup Us$ made up of individuals with positive (Ps), negative(Ns) and uncertain (Us) membership w.r.t. C and a *basic belief assignment* (BBA) m associated with C (with $\Omega = \{+1, -1\}$ as frame of discernment).

The main learning function (see Alg. 1) refines the input test concept using one of the available operators. After candidates are generated, a BBA is computed for each of them. The BBAs for the node concepts are simply estimated based on the number of positive, negative and uncertain instances in the training set:

- $m(\{+1\}) \leftarrow |Ps|/|Ps \cup Ns \cup Us|$;
- $m(\{-1\}) \leftarrow |Ns|/|Ps \cup Ns \cup Us|$;
- $m(\Omega) \leftarrow |Us|/|Ps \cup Ns \cup Us|$;

Example 2 (Computation of a BBA). Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, consider the concepts $\mathtt{Man}$, and its complement $\mathtt{Woman} \equiv \neg\mathtt{Man}$ in $\mathcal{T}$ and the following assertions:
$\{\mathtt{Man(BOB)}, \mathtt{Man(JOHN)}, \mathtt{Woman(ANN)}, \mathtt{Woman(MARY)}\} \subset \mathcal{A}$, with $\mathtt{MARK}$ occurring elsewhere in $\mathcal{A}$ and whose membership w.r.t. $\mathtt{Man}$ is unknown. A BBA for $\mathtt{Man}$ is produced (the frame of discernment $\Omega_{\mathtt{Man}} = \{+1, -1\}$ corresponds to $\{\mathtt{Man}, \mathtt{Woman}\}$):

Algorithm 1. Induction of DST-based TDT

```
1  input: Ps, Ns, Us: set of training instances {positive, negative, uncertain membership}
2          C: concept description,
3          m: BBA
4  output: T: TDT
5  const θ, ν ∈ [0, 1] {thresholds}
6  function INDUCEDSTTDTREE(Ps, Ns, Us:individuals; C:concept; m: BBA)
7  begin
8      if |Ps| = 0 and |Ns| = 0 then
9          begin
10         if Pr⁺ ≥ Pr⁻ then {pre−defined constants wrt the whole training set}
11             T_root ← ⟨C, m⟩
12         else
13             T_root ← ⟨¬C, m⟩
14         return T
15         end
16     if m({−1} = 0 and m({+1}) > θ then
17         begin
18         T_root ← ⟨C, m⟩
19         return T
20         end
21     if m({+1} = 0 and m({−1}) > θ then
22         begin
23         T_root ← ⟨¬C, m⟩
24         return T
25         end
26     if NONSPECIFITY(C) ≥ ν then
27         begin
28         if m({+1}) ≥ m({−1}) then
29             T_root ← ⟨C, m⟩
30         else
31             T_root ← ⟨¬C, m⟩;
32         return T
33         end
34     S ← GENERATECANDIDATES(Ps, Ns, Us)
35     D ← SELECTBESTCONCEPT(m, S)
36     ⟨⟨Pl, Nl, Ul⟩, ⟨Pr, Nr, Ur⟩⟩ ← SPLIT(D, Ps, Ns, Us)
37     T_root ← ⟨D, m_D⟩
38     T_left ← INDUCEDSTTDTREE(Pl, Nl, Ul, D, m_D)
39     T_right ← INDUCEDSTTDTREE(Pr, Nr, Ur, D, m_D)
40     return T
41 end
```

- $m(\{+1\}) = |\{\texttt{BOB}, \texttt{JOHN}\}|/|\{\texttt{BOB}, \texttt{JOHN}, \texttt{ANN}, \texttt{MARY}, \texttt{MARK}\}| = \frac{2}{5} = 0.4$
- $m(\{-1\}) = |\{\texttt{ANN}, \texttt{MARY}\}|/|\{\texttt{BOB}, \texttt{JOHN}, \texttt{ANN}, \texttt{MARY}, \texttt{MARK}\}| = \frac{2}{5} = 0.4$
- $m(\Omega_{\texttt{Man}}) = |\{\texttt{MARK}\}|/|\{\texttt{BOB}, \texttt{JOHN}, \texttt{ANN}, \texttt{MARY}, \texttt{MARK}\}| = \frac{1}{5} = 0.2$ □

The set of candidates S, is made up of pairs $\langle D, m_D\rangle$ where D is a concept description and m_D is the BBA computed for it. After S has been generated, the algorithm selects the best test concept and the corresponding BBA according to measure computed from the BBA and the best pair $\langle D, m_D\rangle$ is installed as a child node of $\langle C, m\rangle$.

This strategy is repeated recursively, splitting the examples according to the test concept in each node. Recursion stops when only positive (resp. negative) instances are rooted to a node which becomes a leaf (see the conditions checked in lines 16 and 21 in Alg. 1). The first condition (line 8) refers to the case when no positive and negative instances reach the node. In this case the algorithm uses priors, Pr^+ and Pr^-, precomputed for the whole training set. The fourth case expresses a situation in which the child nodes added to the tree are characterized by a high non-specificity measure.

Algorithm 2. Candidate test concepts generation

input: Ps, Ns, Us: set of training instances
  C: concept description
output: S: set of $\langle D, m_D \rangle$ {*D is a concept and m is a BBA*}
function GENERATECANDIDATES (Ps, Ns, Us, C): S
begin
$S \leftarrow \emptyset$
$L \leftarrow$ GENERATEREFINEMENTS (C, Ps, Ns) {*based on the refinement operator*}
for each $D \in L$
  begin
  $m \leftarrow$ GETBBA (Ps_D, Ns_D, Us_D)
  $S \leftarrow S \cup \{\langle D, m \rangle\}$
  end
return S
end

Algorithm 3. Selection of the best candidate

input: m_C: BBA,
  S: set of $\langle D, m_D \rangle$
output: C': concept {*best according to non−specificity measure*}
const Ω, Ω_D ($\forall D \in S$): frame of discernment
function SELECTBESTCANDIDATE (m_C, S): concept
begin
  $Ns \leftarrow \sum_{A \subseteq \Omega} m_C(A) \log |A|$
  $C' \leftarrow \text{argmax}_{D \in S} \left(Ns - \sum_{A' \subseteq \Omega_D} m_D(A') \log |A'| \right)$
  return C'
end

For a given concept description D, NONSPECIFICITY(D) is the value computed from its BBA $\sum_{A \in 2^\Omega} m(A) \log(|A|)$ as a measure of imprecision [9]. The algorithm controls the growth by means of the threshold ν. If the condition is verified, the algorithm compares $m(\{+1\})$ and $m(\{-1\})$ to install the proper test concept in the node.

Alg. 2 illustrates how a set S of candidate concepts is generated by GENERATECANDIDATES. This function calls GENERATEREFINEMENTS to generate refinements that can be used as tests. S is updated with pairs $\langle D, m_D \rangle$ where D is a refinement and m_D is the related BBA. Once concepts have been generated, SELECTBESTCANDIDATE (see Alg. 3) selects the best candidate description according to the *non-specificity* measure. The advantage of this method is the explicit representation of the OWA using the maximal ignorance hypothesis (i.e. the one corresponding to Ω).

BBA Creation. As previously mentioned, the proposed approach associates a BBA to each node of a TDT for representing the epistemic uncertainty about the class-membership. The BBA of the child node is created from a subset of the training examples routed to the parent node.

When a branch is created together with the related concept description, the membership of the individuals w.r.t. this concept is computed in order to obtain a BBA whose frame of discernment represents the hypothesis of membership w.r.t. that concept.

Moreover, when a new node is added as left or right child, the algorithm knows about the tests performed on the parent node concept for each instance. Hence, similarly to the Bayesian framework, an implicit kind of conditioning results that allows to relate the membership w.r.t. a concept description in a parent node to the membership w.r.t. the refinements contained in its children.

Stop Criterion. We can apply the DST to TDTs to decide when to stop the growth. As described above, we add a new child node minimizing the degree of imprecision represented by the non-specificity measure. However, if the number of instances with unknown-membership is very high for the new refinement, the imprecision increases and the node should not be further refined. Thus, the algorithm uses two stop criteria:

purity $m(\{-1\}) = 0 \wedge m(\{+1\}) > \theta$ or $m(\{+1\}) = 0 \wedge m(\{-1\}) > \theta$
non-specificity: given a description D, $\textsc{nonSpecificity}(D) > \nu$

The former condition derives from decision tree induction, where a leaf is formed when only instances that belong to a single class remain. In terms of DST, this idea can be represented by a BBA where:

$$\forall A \in 2^{\Omega} \quad m(A) = \begin{cases} 1 & \text{if } A = \{+1\} \text{ (\textit{resp.} } A = \{-1\}) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Thus, the first condition distinguishes two kinds of individuals: those with a certain membership (positive, negative) and those with an uncertain membership. The latter condition moves from the idea that once the best candidate concept description has been chosen, it can be very imprecise (i.e. the measure of non-specificity is very high). Thus, the resulting BBA has the maximum value assigned to a the case of total ignorance w.r.t. the BBA of the parent concept. The threshold ν is used to control this condition.

Prediction. After the TDT has been produced, it can be used to predict the class-membership in the usual way. Given an individual $a \in \mathsf{Ind}(A)$, a path is followed down the tree according to the results of the test w.r.t. the concept D at each node.

Alg. 4 describes the recursive strategy. The base case is when a leaf is reached. In this case, the algorithm updates a list with the BBA associated with the leaf node. The recursive step follows a branch rather than another according to the result of instance-checking w.r.t. the concept description D. If $\mathcal{K} \models D(a)$ the algorithm follows the left branch of the tree. If $\mathcal{K} \models \neg D(a)$ the right branch is followed. A more interesting case occurs when the result of instance check is unknown, i.e. $\mathcal{K} \not\models D(a)$ or $\mathcal{K} \not\models \neg D(a)$. In this case, both the left and the right branch are followed until the leaves are reached. In this way the algorithm can cope with the OWA. The underlying idea is to collect all the possible classifications when the result of a test on an internal node is unknown. In these cases the DST seems to be a good framework in order to combine all such results and make a decision on the membership to be assigned.

After the tree exploration, we may have various BBAs in the list (one per reached leaf). Then, these functions are to be pooled according to a *combination rule* (see Alg. 5). The resulting BBA can be used to compute *belief*, *plausibility* or *confirmation* [5] on the membership hypothesis and the algorithm returns the hypothesis that maximize one of them. Similarly to our previous works [6], we considered the computation of the confirmation in order to balance belief and plausibility in the final decision.

Algorithm 4. Determining the class-membership of an individual

```
input: a: test individual,
       T: TDT,
       K: knowledge base
output: L: list of BBA {related to the leave nodes}
function FINDLEAVES(a, T, K) : L
begin
    N ← ROOT(T)
    if ¬LEAF(N, T) then
        begin
        ⟨D, T_left, T_right⟩ ← INODE(N);
        if K ⊨ D(a) then
            L ← FINDLEAVES(a, T_left, K)
        else if K ⊨ ¬D(a) then
            L ← FINDLEAVES(a, T_right, K)
        else
            begin
            L ← FINDLEAVES(a, T_left, K)
            L ← FINDLEAVES(a, T_right, K)
            end
        end
    else
        begin
        m ← GETBBA(N)
        L ← L ∪ {m}
        end
    return L
end
```

4 Preliminary Experiments

4.1 Setup

The goal of the experimentation is to evaluate the TDTs induced by the proposed method in the class-membership prediction task. We considered various Web ontologies (see Tab. 1). For each of them, 30 query concepts have been randomly generated by combining (using the conjunction and disjunction operators or universal and existential restriction) 2 through 8 concepts of the ontology. A *0.632 bootstrap* procedure was employed for the design of the experiments.

The experiments were repeated under three different conditions. First we ran the original method for learning TDTs. Then, we ran them with the DST-based version

Algorithm 5. Pooling evidence for classification

```
input: a: individual,
       T: TDT,
       K: knowledge base
output: v ⊆ {−1, +1}
function CLASSIFY(a, T, K): v
begin
    L ← FINDLEAVES(a, T, K) {list of BBA}
    m̄ ← ⊕_{m∈L}
    for v ∈ Ω do
        compute \overline{Bel_v} and \overline{Pl_v}
    \overline{Conf_v} ← CONFIRMATION(\overline{Bel_v}, \overline{Pl_v})
    return argmax_{v⊆Ω} \overline{Conf_v}
end
```

Table 1. Ontologies employed in the experiments

Ontology	*Expressiivity*	*Concepts*	*Roles*	*Individuals*
FSM	$\mathcal{SF}(D)$	20	10	37
LEO	$\mathcal{ALCHIF}(D)$	32	26	62
LUBM	$\mathcal{ALEHIF}(D)$	43	25	1555
BIOPAX	$\mathcal{ALCIF}(D)$	74	70	323
NTN	$\mathcal{SHIF}(D)$	47	27	676

with no tree growth control. Finally, we also considered a threshold ($\nu = 0.1$) for non-specificity measure. Higher values would allow for larger trees than those obtained with smaller thresholds.

Due to the disjunctive nature of the concepts represented by the inferred model [2], we have chosen to employ the *Dubois-Prade combination rule* [10] in order to pool BBAs. To compare inductive vs. deductive classification, the following metrics were computed:

- *match*: rate of the test cases (individuals) for which the inductive model and a reasoner predict the same membership (i.e. $+1 \mid +1$, $-1 \mid -1$, $0 \mid 0$);
- *commission*: rate of the cases for which predictions are opposite (i.e. $+1 \mid -1$, $-1 \mid +1$);
- *omission*: rate of test cases for which the inductive method cannot determine a definite membership $(-1, +1)$ while the reasoner is able to do it;
- *induction*: rate of cases where the inductive method can predict a membership while it is not logically derivable.

4.2 Results

Tab. 2 shows the results of the experiments. A low commission rate is noticed for most of the ontologies, except BIOPAX. This rate is higher than the one observed with the standard algorithm. Besides, a low induction rate resulted but in the case of NTN.

In general, the proposed method returns more imprecise results than the results obtained with the TDTs [3], i.e. many times unknown-membership is assigned to test individuals. This is likely due to the combination rule employed. Indeed, the Dubois-Prade combination rule does not take into account the conflict [5]. The pooled BBA is obtained by combining other BBAs considering union of subset of the frame of discernment. Thus, the conflict does not exist and more imprecise results can be obtained. For example,this can occur, when we have two BBAs: the first one has $\{+1\}$ as the only focal element while the other one has $\{-1\}$ as the only focal element. The resulting BBA will have $\{-1, +1\}$ as focal element and an unknown case is returned.

With LUBM this phenomenon is very evident: there are not more induction cases, but the match and the omission rate are very high. In the case of NTN, when the DST-based method is employed, the match rate is lower than with the standard version and

[2] A concept can be obtained easily by visiting the tree and returning the conjunction of the concept descriptions encountered on a path.

Table 2. Results of the experiments using the original terminological trees (DLTree), the DST-TDTs induced with no growth control (DSTTree), and with a growth threshold (DSTG)

Ontology	*Index*	*DLTree*	*DSTTree*	*DSTG*
FSM	*match*	95.34±04.94	93.22±07.33	86.16±10.48
	commiss.	01.81±02.18	01.67±03.05	02.07±03.19
	omission	00.74±02.15	02.57±04.09	04.98±05.99
	induction	02.11±04.42	02.54±01.89	01.16±01.26
LEO	*match*	95.53±10.07	97.07±04.55	94.61±06.75
	commiss.	00.48±00.57	00.41±00.86	00.41±01.00
	omission	03.42±09.84	01.94±04.38	00.58±00.51
	induction	00.57±03.13	00.58±00.51	00.00±00.00
LUBM	*match*	20.78±00.11	79.23±00.11	79.22±00.12
	commiss.	00.00±00.00	00.00±00.00	00.00±00.00
	omission	00.00±00.00	20.77±00.11	20.78±00.12
	induction	79.22±00.11	00.00±00.00	00.00±00.00
BioPax	*match*	96.87±07.35	85.76±21.60	82.15±21.10
	commiss.	01.63±06.44	11.81±19.96	12.32±19.90
	omission	00.30±00.98	01.54±03.02	04.88±03.03
	induction	01.21±00.56	00.89±00.53	00.26±00.27
NTN	*match*	27.02±01.91	18.97±19.01	87.63±00.19
	commiss.	00.00±00.00	00.39±01.08	00.00±00.00
	omission	00.22±00.26	02.09±03.00	12.37±00.19
	induction	72.77±01.51	78.54±17.34	00.00±00.00

commission, omission and induction rate are higher. However, adding a tree-growth control threshold the method shows a more conservative behavior w.r.t. the first experimental condition. In the case of NTN, we observe a lower induction rate and higher match and omission rates. Instead, the commission error rate is lower. The increase of the induction rate and the decrease of the match rate for the DST-based method (without the tree-growth control threshold) are likely due to uncertain membership cases for which the algorithm can determine the class.

A final remark regards the stability for the proposed method: the outcomes show a higher standard deviation w.r.t. the original version, hence it seems less stable so far.

5 Conclusions and Extensions

In this work a novel type of terminological decision trees and the related learning algorithms have been proposed, in order to integrate forms of epistemic uncertainty in such an inductive classification model. We have shown how the DST can be employed together with machine learning methods for the Semantic Web representations as an alternative framework to cope with the inherent uncertainty and incompleteness. The proposed algorithm can discover potentially new (non logically derivable) assertions that can be used to complete the extensional part of a Web ontology (or a Linked Data dataset) whose expressiveness allows to represent concepts by means of disjunction and complement operators. However, experimental results show that the current version of the method may have sometimes a worse performance, especially in terms of match rate and stability.

The proposed method can be extended along various directions. It is possible to use a total uncertainty measure that integrates conflicting evidence [9]. In the current version, we control the growth of the tree. A further extension may concern the definition of a pruning method based on the DST.

Acknowledgments. This work fulfils the research objectives of the PON 02_00563_34 70993 project "VINCENTE - A Virtual collective INtelligenCe ENvironment to develop sustainable Technology Entrepreneurship ecosystems" and the PON 02_00563 _3489339 project "PUGLIA@SERVICE - Internet-based Service Engineering enabling Smart Territory structural development" funded by the Italian Ministry of University and Research (MIUR).

References

1. Rettinger, A., Lösch, U., Tresp, V., d'Amato, C., Fanizzi, N.: Mining the semantic web - statistical learning for next generation knowledge bases. Data Min. Knowl. Discov. 24, 613–662 (2012)
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook, 2nd edn. Cambridge University Press (2007)
3. Fanizzi, N., d'Amato, C., Esposito, F.: Induction of concepts in web ontologies through terminological decision trees. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010, Part I. LNCS, vol. 6321, pp. 442–457. Springer, Heidelberg (2010)
4. Blockeel, H., De Raedt, L.: Top-down induction of first-order logical decision trees. Artif. Intell. 101, 285–297 (1998)
5. Klir, J.: Uncertainty and Information. Wiley (2006)
6. Rizzo, G., d'Amato, C., Fanizzi, N., Esposito, F.: Assertion prediction with ontologies through evidence combination. In: Bobillo, F., Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) URSW 2008-2010/UniDL 2010. LNCS, vol. 7123, pp. 282–299. Springer, Heidelberg (2013)
7. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers (2011)
8. Denoeux, T.: A k-nearest neighbor classification rule based on Dempster-Shafer theory. IEEE Transactions on Systems, Man and Cybernetics 25, 804–813 (1995)
9. Sutton-Charani, N., Destercke, S., Denœux, T.: Classification trees based on belief functions. In: Denœux, T., Masson, M.-H. (eds.) Belief Functions: Theory & Appl. AISC, vol. 164, pp. 77–84. Springer, Heidelberg (2012)
10. Dubois, D., Prade, H.: On the combination of evidence in various mathematical frameworks. In: Flamm, J., Luisi, T. (eds.) Reliability Data Collection and Analysis. Eurocourses, vol. 3, pp. 213–241. Springer Netherlands (1992)