Noname manuscript No.
(will be inserted by the editor)

Mining the Semantic Web

Statistical Learning for Next Generation Knowledge Bases

Achim Rettinger · Uta Lösch · Volker Tresp · Claudia d'Amato · Nicola Fanizzi

the date of receipt and acceptance should be inserted later

Abstract In the Semantic Web vision of the World Wide Web, content will not only be accessible to humans but will also be available in machine interpretable form as ontological knowledge bases. Ontological knowledge bases enable formal querying and reasoning and, consequently, a main research focus has been the investigation of how deductive reasoning can be utilized in ontological representations to enable more advanced applications.

However, purely logic methods have not yet proven to be very effective for several reasons: First, there still is the unsolved problem of scalability of reasoning to Web scale. Second, logical reasoning has problems with uncertain information, which is abundant on Semantic Web data due to its distributed and heterogeneous nature. Third, the construction of ontological knowledge bases suitable for advanced reasoning techniques is complex, which ultimately results in a lack of such expressive real-world data sets with large amounts of instance data.

From another perspective, the more expressive structured representations open up new opportunities for data mining, knowledge extraction and machine learning techniques. If moving towards the idea that part of the knowledge already lies in the data, inductive methods appear promising, in particular since inductive methods can inherently handle noisy, inconsistent, uncertain and missing data. While there has been broad coverage of inducing concept structures from less structured sources (text, Web pages), like in ontology learning, given the problems mentioned above, we focus on new methods for dealing with Semantic Web knowledge bases, relying on statistical inference on their standard representations.

We argue that machine learning research has to offer a wide variety of methods applicable to different expressivity levels of Semantic Web knowledge bases: Ranging from weakly expressive but widely available knowledge bases in RDF to highly expressive first-order

A. Rettinger · U. Lösch
Institute AIFB, Karlsruhe Institute of Technology, KIT-Campus Süd, 76128 Karlsruhe, Germany
E-mail: {rettinger, uta.loesch}@kit.edu

V. Tresp
Siemens Corporate Technologies, Otto-Hahn-Ring 6, 81739 Munich, Germany
E-mail: volker.tresp@siemens.com

C. d'Amato · N. Fanizzi
Dipartimento di informatica, Università degli studi di Bari "Aldo Moro", Campus Univ., Via Orabona 4, 70125 Bari, Italy
E-mail: {claudia.damato, fanizzi}@di.uniba.it

knowledge bases, this paper surveys statistical approaches to mining the Semantic Web. We specifically cover similarity and distance-based methods, kernel machines, multivariate prediction models, relational graphical models and first-order probabilistic learning approaches and discuss their applicability to Semantic Web representations. Finally, we present selected experiments which were conducted on Semantic Web mining tasks for some of the algorithms presented before. This is intended to show the breadth and general potential of this exiting new research and application area for data mining.

Keywords: Semantic Web, Ontology, Knowledge Representation, Description Logics, RDF, Linked Data, Semantic Similarity, Kernels, Multivariate Prediction, First-order Probabilistic Learning, Relational Graphical Models.

1 Introduction

The World Wide Web is mostly accessible to humans whereas machines only have a very rudimentary *understanding* of its content. The original vision behind the Semantic Web (henceforth SW) is that computers should somehow be able to *understand* and exploit information offered on the Web (Berners-Lee et al., 2001). In the near future, a Web representation may contain both human-readable and machine-interpretable sections accessible for automated processing. The vast amount of linked data becoming available in recent years is one clear indication for this trend (Bizer et al., 2009).

The SW is based on two components: 1) Formal ontologies provide domain-specific background knowledge as a vocabulary that is shared by several parties and that describes abstract object classes, predicate classes and their interdependencies, formalized in logical statements; 2) Annotations of web resources with statements which can be read and interpreted by machines via the common ontological knowledge present instantiated real world observations.

SW technologies will form the infrastructure for a standardized representation of information and for information exchange. Reasoning is expected to play an important role in this context: Based on ontological knowledge and an initial set of statements, reasoning can derive implicit statements by deductive inference. However, logical reasoning has its limitations. First, it does not easily scale up to the size of the Web as required by many applications. Second, uncertain information is originally not considered in the design of the SW and this issue has only recently been addressed (da Costa et al., 2008). Third, reasoning is completely based on axiomatic prior knowledge and does not exploit regularities in the data that have not been formulated as ontological knowledge. Fourth, knowledge bases suitable for expressive reasoning tasks require an expensive construction process ultimately resulting in a lack of real world data sets which fulfill the formal requirements.

In contrast, as demonstrated in many application areas, successful solutions can often be achieved by the inductive approach used in machine learning (ML) and data mining. In a way, inductive methods perform approximate reasoning and derive predictions which are neither explicitly asserted in the knowledge base nor provable (resp. refutable) based on logical reasoning: e.g., the reasoning task of *classification* which assigns instances to entity classes can be stated as a *multi-label prediction* task where each instance is to be assigned to a subset of the possible class labels.

In this paper, we discuss existing applications of machine learning methods to SW data. We focus on algorithms that are suitable for the relational character of the SW data structure, particular aspects that are likely to be relevant for the SW such as scalability, missing and contradicting data, and the potential to integrate ontological background knowledge.

So far, in the context of the SW, machine learning has been mostly considered as a tool to enrich or extend ontologies on the schema level. More precisely, ML may serve the SW by supporting ontology *construction* and *management*, ontology *evaluation*, ontology *refinement*, ontology *evolution*, as well as the *mapping*, *merging* and *alignment* of ontologies (Bloehdorn et al., 2006; Euzenat and Shvaiko, 2007; Grobelnik and Mladenic, 2006; Maedche and Staab, 2004). Another task is learning logical constraints formulated in the language of the employed ontology (Cohen and Hirsh, 1994; Fanizzi et al., 2008a; Iannone et al., 2007; Lehmann and Hitzler, 2008; Lehmann, 2009; Lisi and Esposito, 2005). In all these tasks, machine learning needs to produce deterministic logical statements by using, e.g., methods from inductive logic programming. ML methods have also been applied in ontology *learning* (Buitelaar et al., 2004; Cimiano et al., 2005; Cimiano and Völker, 2005; Velardi et al., 2005; Poon and Domingos, 2010). Probabilistic variants of this task have been termed *structure learning* in statistical relation learning (Huynh and Mooney, 2011).

While there the problem consists in deriving ontological knowledge from external data sources such as text collections, we are concerned with methods which induce instance knowledge based on the knowledge representation itself. The induced statistical models can be used to estimate the probability that statements on the instance level are true, which are neither explicitly asserted in the database nor can be proven (or refuted) based on logical reasoning. This can also be considered as an enhancement of traditional Web mining to *SW mining* by applying ML to the SW (Stumme et al., 2006).

Note that this survey is different from other surveys published in the context of the SW in that it discusses the application of ML methods to data generated in this field. In contrast, Ding et al. (2007) present an overview of SW standards themselves and sample applications of ontologies in different scenarios; Nixon et al. (2008) discuss the issues arising from the communication and coordination which are specific to SW data; Tiropanis et al. (2009) give an overview of SW tools which can be employed in the areas of digital libraries, e-learning and virtual communities.

Challenges for Mining the SW: Applying ML to the SW demands traditional ML techniques to be adapted to the specific properties of ontologies. For instance, statistical learning on the SW should be highly scalable because of the very large number of statements. Query processing, which is essential for the calculation of the features, should be executed efficiently (Baader et al., 2003).

Another important issue in SW learning concerns missing or incomplete data. One cannot make a closed-world assumption and postulate that the only true facts are those asserted in (or provable from) the current knowledge base. Following the open-world assumption made in the SW context, one should assume that the truth values of unspecified and not derivable statements are *unknown*. Unfortunately, missing at random mechanisms are typically not applicable since the SW data typically only contain statements on known true statements and not on known false statements.

Finally, there are approaches that strive towards a complete theoretical treatment of expressive ontology constructs (see Sec. 7) by extending the logical formalism to being able to handle uncertainty and thus truly combine deductive and inductive reasoning.

Content of this article: The SW offers great potential for the application of ML algorithms on two levels: First, large real-world SW data sets in weakly expressive data representations (like linked data (Bizer et al., 2009)) are becoming available. Second, in the future it is expected that content on the web will be made available using highly expressive data representations (like description logic (Baader et al., 2003)). In this paper we intend to survey

existing ML algorithms that can be applied to both levels of expressivity of SW knowledge representations. As machine learning and data mining on the SW is still in its infancy there is only little work on concrete applications, i.e., on empirical testing and on the evaluation of proposed ML techniques based on SW data. Thus, to demonstrate typical SW data mining tasks and the applicability of such algorithms we present new experiments and summarize selected experiments published previously (cmp. Sec. 8 and Huang et al. (2010); Rettinger et al. (2009)).

In the remaining part of the paper, we concentrate on a survey of the statistical learning models and related techniques already proposed or potentially suited for SW data representations (Sec. 3-7). But first, in Sec. 2, we briefly introduce SW knowledge representations that are the input for potential ML algorithms. We associate the representations with suitable ML tasks which will later be applied to real world data sets in Sec. 8. Starting from Sec. 3, we discuss different learning algorithms suited for SW representations. In the first part, we present instance-based methods exploiting semantic similarity measures which extend the nearest neighbor approach (Sec. 3) to more complex models, such as multilayer networks (Sec. 3.3) and kernel machines (Sec. 4). The second part (Sec. 5, Sec. 6 and Sec. 7) focuses on the latest developments IN statistical learning with relational representations, namely matrix/tensor decomposition, relational graphical models and first-order probabilistic approaches. This is also known as Statistical Relational Learning (SRL) (Getoor and Taskar, 2007) which we examine in the context of performing probabilistic inference on SW representations.

Finally, we give a selection of experimental evaluation of some of the algorithms, described in the preceding sections, to real-world SW data sources (see Sec. 8) before we conclude (see Sec. 9).

2 Semantic Web Knowledge Bases and Related Machine Learning Tasks

The main standard[1] representations for knowledge bases in the Semantic Web are *RDF (Resource Description Framework)*, *RDFS (Resource Description Framework Schema)* and *OWL (Web Ontology Language)*. RDF is used for specifying statements about instances and RDFS defines schema and subclass hierarchies. OWL can be used to extend RDFS to formulate more expressive schema and subclass hierarchies and additional logical constraints. The statements in RDF, RDFS and OWL can all be represented as one combined directed graph. A common semantics relies on the predefined domain-independent interpretations of the RDFS and OWL components.

Note that RDF and RDFS build the backbone of today's semantic web. Most SW data available today is defined using these lightweight formalisms. While OWL allows for more expressive formalizations and thus for more powerful inferencing tasks, no data sets WITH A LARGE NUMBER OF INSTANCES AND HIGHLY EXPRESSIVE ONTOLOGICAL CONSTRUCTS ARE CURRENTLY available. This also limits the need, development and testing of ML-techniques for OWL.

In this Section we intend to introduce RDF(S) (presented in Subsection 2.1) and OWL (presented in Subsection 2.2) using simple examples accompanied by potential ML tasks to be performed on such data sets. We conclude this section with proposed probabilistic extensions to SW knowledge bases (see Subsection 2.3) which would seamlessly integrate

[1] See references in the SW section of the W3C website: `http://www.w3.org/standards/semanticweb/`

probabilistic reasoning with SW knowledge bases. However, the focus of the remaining paper is on ML techniques for the established formalisms in Subsection 2.1 and 2.2.

2.1 Resource Description Framework (RDF)

Data Representation: RDF defines the data model for the SW. It has been developed to represent information about Web resources, uniquely identified via a URI (Unique Ressource Identifier). The basic statement is a triple of the form `(subject, property, property value)` or, equivalently, `(subject, predicate, object)`. The subject of a triple is a resource, which is identified by a URI. The predicate is also denoted by a URI, and the object is either another resource or datatype value, also called literal. In case the property value is a resource the property is called an *object property*, otherwise it is called *datatype property*.

Example 1 RDF allows us to encode basic information about persons (using RDF abstract syntax)[2]. In the example two persons are defined whose names are John Doe respectively Jane Doe and who are both interested in the topic Machine Learning:

```
person100     foaf:name            "John Doe"
person100     foaf:topic_interest  topic110
topic110      skos:prefLabel       "Machine Learning"
person100     foaf:knows           person200
person200     foaf:name            "Jane Doe"
person200     foaf:topic_interest  topic110
person100     rdf:type             foaf:Person
person200     rdf:type             foaf:Person
```

A RDF knowledge base consists of a set of triples which can be represented as a directed graph. All elements that occur either as subject or object of a triple are vertices in the graph and each triple (s, p, o) defines an edge which goes from s to o and has label p. Thus, each triple expresses a binary relation. RDF also allows for the expression of n-ary relations by means of special blank nodes which can not be referenced using a URI and which serve as a link between the n elements involved in the relation.

RDF also allows for the definition of type-relations, which associate a resource with a special resource denoting a concept. Each resource may be associated with one or several concepts (i.e. classes) via the type-property. Concepts represent the set of all instances which are stated as belonging to that concept. RDF-Schema (or RDFS) allows for the definition of restrictions on properties and concepts. The most important of these restrictions are subclass relationships and domains and ranges of properties. Simple rule-based entailments are possible on the combined RDF/RDFS graph.

Example 2 RDFS allows for the encoding of restrictions on the properties and the concepts in the data graph. Here, we express that the `foaf:name` property links a person to a literal:

```
foaf:name     rdfs:domain          foaf:Person
foaf:name     rdfs:range           rdfs:Literal
```

[2] The statements use property names taken from well-known RDF-based metadata standards such as FOAF (Brickley and Miller, 2007) and SKOS (Miles and Brickley, 2005), marked by the corresponding namespaces.

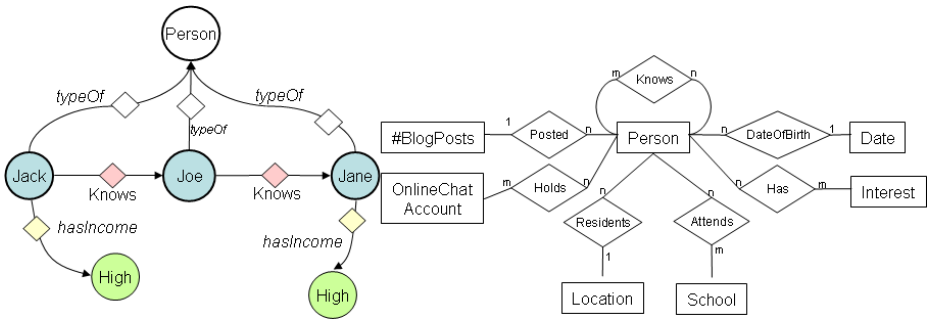

Fig. 1 Left: Example of an RDF graph displaying a social friendship network in which the income of a person is an attribute. Resources are represented by circular nodes and triples ARE represented by labeled directed from subject node to object node. The diamond-shaped nodes represent random variables which are in state one if the corresponding triple exists. Nodes representing statistical units (in both cases: *Persons*) have a darker rim. Right: Entity-relationship diagram of the LJ-FOAF domain. Note, not all constructs of RDFS can be transformed to representations suitable to existing ML algorithms and with similar semantics. This is especially true for more expressive representations like DL and OWL, as introduced in the next section. .

Machine Learning tasks: An RDF data set can be seen as a node and edge labeled graph where each node label appears only once but often with a high degree. However, the semantics added by RDFS can not be expressed with standard graph theoretic terminology. The same applies to the mapping of RDF and RDFS to relational databases. E.g., hierarchical data is not part of a standard relational data base. An example of an entity-relationship diagram extracted from an RDF graph is given in Fig. 1.

Still, as RDF data sources can – to some extent – be mapped to graph structures or relational data bases, many ML tasks on graphs or data bases apply to RDF as well. On RDF instances, variants of classical ML tasks, like classification, can be performed. This could be the assignment of individuals to classes, like `person`-instances to `foaf:person`-classes if this information is only partially known. COMMENT: THE LAST SENTENCE MAKES NO SENSE TO ME. To avoid confusion with the ML terminology we will use *class-membership prediction* for this task (sometimes also called *object type prediction* (Getoor et al., 2007)) and present experiments in Sec. 8.1.1. Another classification task is the prediction of features of ENTITIES, like `foaf:topic_interest` of a person, which we will call *property value prediction* and present experiments in Sec. 8.1.2. An important task that has been of increasing interest is *relation* or *link prediction*. In our example this could be `foaf:knows` recommendations to persons (for experiments see Sec. 8.2). Another important conventional data mining task applicable to RDF data is the *clustering* of instances like similar persons which we present in Sec. 8.3 (sometimes also called *group detection* (Getoor et al., 2007)). A potential task which could be performed on RDF instance data, but to the best of our knowledge has not been tried yet on RDF sources is *link classification* where a relation instance is assigned to a given hierarchy of relation classes (Getoor et al., 2007). A related task not covered here is *entity resolution* where the goal is to detect identical instances occurring more than once in a SW data source (Singla and Domingos, 2006).

2.2 Description Logics and Web Ontology Language (OWL)

Data representation: Description Logics (DLs) are a family of knowledge representation formalisms whose design is especially focused on the careful formalization of knowledge

and on precisely defined reasoning techniques (see the handbook (Baader et al., 2003) for a thorough reference). They can be used to model an application domain by defining its relevant concepts (the classes) and specifying properties of objects and individuals (resources) contained in the knowledge bases. The formal definition of the interpretation of individuals, concepts and relationships is given by a model-theoretic semantics which allows for many inference services which are concretely provided by reasoners.

The Web Ontology Language (OWL) in its variant OWL DL is a specific *Description Logic* (DL) (Baader et al., 2003).

Formally, a DL *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ contains a *TBox* $\mathcal{T}$ and an *ABox* $\mathcal{A}$. $\mathcal{T}$ is a set of (equivalence or inclusion) axioms that define concepts (sets of individuals) and roles (binary relationships) using a vocabulary of primitive (atomic) concept and role names and combining them through specific language constructors. $\mathcal{A}$ contains factual assertions concerning the individuals. Depending on the expressivity of the specific DL, different language constructors are available.

OWL DL enables stating the equivalence or disjointness of classes and the (non-)identity of properties respectively instances. The behavior of properties can be classified as being symmetric, transitive, functional or inverse functional. OWL DL allows the user to construct classes by enumerating their content, through forming intersections, unions and complements of classes. Also classes can be defined via property restrictions. Furthermore, cardinality constraints can be formulated.

Example 3 To continue the above example, let us define the concept of machine learning fans as persons who are interested in Machine Learning:

$$\texttt{MachineLearningFan} \equiv \texttt{foaf:Person} \sqcap \exists \texttt{foaf:topic\_interest.topic110}$$

Using the knowledge base from the RDF example and this axiom a reasoner can deduce that Jane Doe and John Doe are Machine Learning fans. Additionally, the fact that the class of Machine Learning fans is a subclass of the class foaf:Person can be inferred.

The various DL variants are characterized by a definite set of constuctors for complex concepts (resp. roles). They constitute specific fragments of First Order Logic (FOL) or even higher order extensions. Therefore they inherit a standard (set-theoretic) semantics for the axioms/assertions in terms of interpretations for the concept/role names, individuals and the complex descriptions that can be built by means of the available operators.

Reasoning is based on the interpretations that satisfy the axioms/assertions in the knowledge base (i.e. their logic *models*). Hence the standard notions of *satisfiability*, *validity* and *entailment* (denoted with $\models$) defined for FOL naturally extend to DLs. However, it is worthwhile to point out that an open-world assumption (OWA) is made in Description Logics, which is quite convenient for large scale application scenarios. Hence these representations and their related inference services differ from the standard settings adopted with databases (and logic theories/programs) which are characterized by a closed-world semantics. Reasoning with DLs requires more than (finite) model checking or query answering with DBs: generally it is performed through tableau-based procedures (see (Baader et al., 2003)).

In OWL DL the *unique names assumption* is made on the individual names. This turns out to be handy for performing basic operations which may be crucial in statistical methods, such as counting. They are concretely represented in the OWL syntax by their distinctive URIs. The set of the distinct individuals occurring in the *ABox* $\mathcal{A}$ is generally denoted with $\mathsf{Ind}(\mathcal{A})$.

One of the most useful inferences on individuals required by the inductive methods is *instance-checking* (Baader et al., 2003), which amounts to determining whether an individual, say a, is an instance of some concept C with respect to the models of the knowledge base (we call the inductive version of this task *class assignment*). This is denoted by $\mathcal{K} \models C(a)$. The dual inference service, named *retrieval*, amounts to finding the individuals that belong to a certain concept C, i.e. finding the set $\mathsf{retrieval}_{\mathcal{K}}(C) = \{a \in \mathsf{Ind}(\mathcal{A}) \mid \mathcal{K} \models C(a)\}$. Services like those are provided by DL reasoners. Note that due to the open-world assumption, a reasoner may not be able to give a positive or negative answer to such membership queries, as both models in which the membership relation holds and models in which it is refuted can be constructed.

Machine Learning tasks: In the context of more expressive DL ontologies which allow more refined operators for specifying knowledge one could also think about several ML tasks on the structural level or – using the DL terminology – *TBox* $\mathcal{T}$. This could e.g. be *link cardinality estimation* or *predicate invention* where one tries to induce the cardinality of relation classes or even define new relation classes both from existing link instances (Getoor et al., 2007). However, as we focus only on the instance level, I.E., – USING THE DL TERMINOLOGY – THE *ABox* $\mathcal{A}$, all the additional operators basically impose restrictions on potential individuals in the knowledge base. This use of ontologies as background knowledge in the learning task is an interesting aspect (see Section 8.2.2). As more expressive reasoning tasks can be solved with DL, the use of inductive methods in the application to reasoning tasks becomes an interesting option (see Section 8.1.1) as well.

2.3 Probabilistic Extensions to SW Knowledge Bases

The formal ontology languages we have discussed so far are deterministic and not intended to describe or infer uncertain knowledge extracted via ML. There are a number of proposals for extending logical languages with probabilistic information. Here we will only consider approaches that directly extend SW languages, in particular RDF and OWL DL, and DL in general. We will adopt the classification of (Predoiu and Stuckenschmidt, 2008).

First, there are models that build on the syntax of established semantic web languages, examples being pRDF and BayesOWL (Ding, 2005). pRDF is a probabilistic extension to RDF which represents the different elements of a Bayesian network and links them to regular RDF statements. In essence, RDF is used as the language to describe a Bayesian network. Inference capabilities equal those of Bayesian networks and RDF(S) inference cannot be integrated. BayesOWL is a probabilistic extension of OWL. It represents probabilistic information about class membership within OWL ontologies. The main reasoning task that can be performed in BayesOWL is to calculate the membership probability of an instance for all the classes in the ontology. Both pRDF and BayesOWL support probabilistic reasoning for only a small subset of the expressiveness of the languages which they extend.

Second, there are formalisms that consider extensions of Description Logics, examples being P-$\mathcal{SHOQ}(D)$ (Giugno and Lukasiewicz, 2002) and P-CLASSIC (Koller et al., 1997). P-CLASSIC is a probabilistic extension of the CLASSIC Description Logics which adds probabilities to roles of typical instances by the use of a Bayesian network. Reasoning tasks in P-CLASSIC compute the probability of a complex concept expression based on the definition of the joint probability distribution over atomic classes and features of relations. Again, only the reasoning capabilities of Bayesian networks can be used. As for P-$\mathcal{SHOQ}(D)$ no reasoning tools have been devised. However, the reasoning tasks that can potentially be

solved are quite powerful e. g. , it can be determined whether a given knowledge base is consistent.

These two examples are probabilistic extensions of DL. The third extension considers approaches that integrate probabilistic Logic Programming variants with Description Logic. Such extensions are e. g. , Bayesian Description Logic Programs (Predoiu, 2006) or Probabilistic Description Logic Programs (Lukasiewicz, 2007). In one version, OWL is integrated in Logic Programming by specifying a logic program and a description logics knowledge base. In a second version, OWL is integrated in Logic Programming formalisms that have been extended with probabilities.

3 Instance-Based Learning for Class-Membership Prediction

At the instance level, the most widely used inference services for ontological knowledge are *instance checking* and *retrieval* which consist respectively in assessing if an individual is instance of a given concept and in determining all individuals that are instances of a given concept (see Sect. 2.2). Generally, these services are provided by resorting to standard deductive reasoning procedures (Baader et al., 2003). However, logic reasoning may be both too demanding, because of its complexity, and also error-prone because of accidental inconsistency or (inherent) incompleteness in the knowledge bases that are fed by heterogeneous and distributed data sources.

One path of research pursues approximate reasoning procedures (Hitzler and Vrandečić, 2005). Another one, and the approach discussed in this paper, is based on instance-based learning, by casting the class assignment problem to predicting the classes (i.e. the concepts) of an individual. Efficient learning methods, originally developed for simple representations (attribute vectors derived from propositional logic), can be effectively upgraded to work with richer structured representations (Gärtner et al., 2004), including the standard ontology languages.

Similarity-based methods have been shown to effectively solve *supervised* and *unsupervised* learning problems in standard representations for the Semantic Web (d'Amato et al., 2008b; Fanizzi et al., 2008b). In particular, an inductive model based on similarity functions for individuals within ontological knowledge bases can be adopted for efficiently predicting the class-membership of further individuals w.r.t. a given target concept. Note, that in contrast to the usual *closed-world* semantics of query answering from databases, the OWA is made in the context of the DL knowledge bases. Therefore, a ternary value set $V = \{+1, -1, 0\}$ is considered for indicating, membership, non-membership and uncertain membership assignment w.r.t. the given target concept Q, respectively.

Prediction is performed on the grounds of the hypotheses (decision functions) induced from a set of training examples whose correct labels are provided by an expert. Note that, in this specific setting, hypotheses for both membership and non-membership may have to be learned for each concept. Prediction is meant to be performed very efficiently especially in comparison to the complexity of reasoning with expressive DL languages (see (Baader et al., 2003), ch. 3). Moreover, the classifier may be able, in some cases, to determine the membership w.r.t. Q (resp. $\neg Q$) even when the reasoner cannot (due to the OWA). This means that an inductive procedure may be able to predict assertions that are likely to hold but it is not logically derivable.

Dually, class-membership prediction can be used to provide an approximate concept retrieval service. This learning problem can be defined as follows: given a (limited) set of individuals that are examples and counterexamples for the target query concept Q (i.e.

instances of Q and $\neg Q$, resp.), using a learning algorithm, induce hypotheses (e.g. decision functions) that can be exploited to select the instances of Q among the individuals in $\mathsf{Ind}(\mathcal{A})$.

3.1 Similarity and Distance between Individuals

In order to apply similarity-based learning methods to the standard representations for the Semantic Web a notion of (dis-)similarity is necessary. Various attempts to define semantic similarity (or dissimilarity) measures for concept languages have been made. Among them, two main approaches can be distinguished: 1) measures based on semantic relations (also called path distance measures); 2) measures based on concept extensions and *Information Content*.

In the former approach all concepts are in an *is-a* taxonomy, and the similarity between two concepts is computed by counting the (weighted) edges in the paths from the considered concepts to their most specific ancestor. Concepts with a few links separating them are similar; concepts with many links between them are less similar (Rada et al., 1989; Lee et al., 1993; Bright et al., 1994; Maynard et al., 2006). In the latter approach the similarity value is computed by counting the common instances of the concept extensions (d'Amato et al., 2005) or by measuring the variation of the *Information Content* between the considered concepts (d'Amato et al., 2006a; Resnik, 1999; Borgida et al., 2005).

Since the ontology does not have the simple structure of a taxonomy, but it is rather an elaborated graph, similarity measures based on path distances cannot be used. The measures based on overlap of concept extensions can be more easily applied to DL representation, however they are not able to capture the similarities between disjoint concepts (d'Amato et al., 2008c). For this reason, measures that adopt a miscellaneous approach, namely structural and extensional-based, have been defined (d'Amato et al., 2006a; Janowicz, 2006; Janowicz et al., 2007; d'Amato et al., 2008c). The main drawback of these measures is that they hardly scale to high expressive DLs.

Furthermore, for the intended purposes, a function is required for measuring the similarity of individuals rather than concepts. It can be observed that individuals do not have a syntactic structure that can be compared. A solution is to lift them to the concept description level before comparing them (d'Amato et al., 2005, 2006a, 2008c) (recurring to the notion of the *most specific concept* of an individual w.r.t. the ABox (Baader et al., 2003)). Yet this makes the measure language-dependent. Besides, it would add a further approximation as the most specific concepts can be defined only for simple DLs. An alternative approach has been proposed in (Janowicz and Wilkes, 2009) where individuals are compared by generating concept models as the usual approach adopted by the tableaux reasoning algorithm. However, in this case, the measure could be hardly applicable in presence of inconsistencies.

The inductive procedures to be surveyed in the following exploit measures that totally depend on semantic aspects of the individuals in the knowledge base. The similarity should be measured by specific metrics which should be sensible to the semantics of the individuals.

Following some ideas borrowed from (Sebag, 1997), totally semantic distance measures for individuals can be defined in the context of a DL knowledge base independently of the underlying DL language (Fanizzi et al., 2007). These measures are based on the idea of comparing the semantics of the input individuals along a number of dimensions (acting as context for individuals) represented by a committee of concept descriptions. Indeed, on a semantic level, similar individuals should behave similarly with respect to the same concepts. More formally, the rationale is to compare individuals on the grounds of their semantics w.r.t. a collection of concept descriptions F, which stands as a group of discriminating *features*

expressed in the DL language of choice. The family of distance functions for individuals inspired by Minkowski's norms L_p can be defined as follows (Fanizzi et al., 2007; d'Amato et al., 2008b). Given a set of concept descriptions $\mathsf{F} = \{F_1, F_2, \ldots, F_m\}$ and a weight vector $\mathbf{w}$, a family of dissimilarity measures $d_p^{\mathsf{F}} : \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mapsto [0, 1]$, with $p > 0$, is defined as follows:

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

where $p > 0$ and $\forall i \in \{1, \ldots, m\}$ the function δ_i is defined by:

$$\forall (a, b) \in (\mathsf{Ind}(\mathcal{A}))^2 \qquad \delta_i(a, b) = \begin{cases} 0 & (\mathcal{K} \models F_i(a) \wedge \mathcal{K} \models F_i(b)) \vee (\mathcal{K} \models \neg F_i(a) \wedge \mathcal{K} \models \neg F_i(b)) \\ 1 & (\mathcal{K} \models F_i(a) \wedge \mathcal{K} \models \neg F_i(b)) \vee (\mathcal{K} \models \neg F_i(a) \wedge \mathcal{K} \models F_i(b)) \\ u_i & otherwise \end{cases}$$

With densely populated knowledge bases, it may be possible to approximate the tests $\mathcal{K} \models F_i(x)$ with simple ABox lookups $F_i(x) \in \mathcal{A}$ which are less computationally demanding. Likely, this definition of the δ_i is also less prone to accidental errors, since noisy assertions should be considered as rare. Of course the effectiveness depends on the density of the assertions available for the various feature concepts. Even more so, when a KBMS is employed such as instance stores, such information may be maintained and be ready-to-use (at least for the concepts in F).

In lack of prior knowledge, the features in F are assumed to have uniform weights (encoded in the weight vector $\mathbf{w}$). Alternatively, such features could be weighted in order to reflect the impact of the single feature concept with respect to the overall dissimilarity (see discussion below).

The value u_i, generally considered as $1/2$, can be fine-tuned to reflect the degree of uncertainty on the membership w.r.t. the given feature, e. g. , $u_i \approx \Pr(\mathcal{K} \not\models F_i(a)) + \Pr(\mathcal{K} \not\models F_i(b)) - \Pr(\mathcal{K} \not\models F_i(a)) \cdot \Pr(\mathcal{K} \not\models F_i(b))$.

It is easy to prove (Fanizzi et al., 2007) that the d_p^{F} are pseudo-metrics (i.e. semi-distances (Bock, 1999)).

Compared to other dissimilarity measures proposed in the literature (Borgida et al., 2005; d'Amato et al., 2008a; Janowicz, 2006), the presented functions do not depend on the constructors of a specific language, rather they require only (retrieval or) instance-checking for computing the δ_i's through concept-membership queries to the knowledge base.

The measures strongly depend on the feature set F. Here, the assumption made is that F represents a sufficient number of (possibly redundant) features that are able to discriminate really different individuals. The choice of the concepts to be included – *feature selection* – is beyond the scope of this paper. See (Fanizzi et al., 2007, 2008b) for randomized optimization procedures aimed at finding optimal committees. Experimentally, good results were observed by using the very set of both primitive and defined concepts found in the knowledge base (d'Amato et al., 2008b).

Note that the original measures (Fanizzi et al., 2007) correspond to the case of a vector $\mathbf{w}$ of *uniform* weights. However the single components w_i should reflect the impact of the various feature concepts w.r.t. the overall similarity. For this reason, different choices can be made depending on two discernibility criteria.

Entropy-based Weights The *information* conveyed by a feature can be estimated by its entropy (d'Amato et al., 2008b). Namely, the extension of a feature F_i with respect to the

whole domain of objects may be probabilistically quantified as $P_i^+ = |F_i^{\mathcal{I}}|/|\Delta^{\mathcal{I}}|$ (w.r.t. the canonical interpretation $\mathcal{I}$ whose domain is made up by the individual names occurring in the ABox (Baader et al., 2003)). This can be roughly approximated with: $P_i^+ = |\mathsf{retrieval}(F_i)|/|\mathsf{Ind}(\mathcal{A})|$, where $\mathsf{retrieval}$ denotes the result of the DL instance retrieval service (Baader et al., 2003). Hence, considering also the probability P_i^- related to its negation and that related to the individuals with uncertain membership (w.r.t. F_i), $P_i^U = 1 - (P_i^+ + P_i^-)$, one may determine an entropic measure for the discernibility yielded by the feature: $h_i = -P_i \log(P_i) - P_i^- \log(P_i^-) - P_i^U \log(P_i^U)$. The weights may be derived by normalization $w_i = h_i/||\mathbf{h}||$.

Variance-based Weights An alternative is based on an estimate of the *feature variance*. Following the method proposed in (Hastie et al., 2001), the estimate can be defined as follows: $\widehat{\mathrm{var}}(F_i) = \sum_{a \in \mathsf{Ind}(\mathcal{A})} \sum_{b \in \mathsf{Ind}(\mathcal{A})} [\pi_i(a) - \pi_i(b)]^2$, where, for $i = 1, \ldots, m$, the *projection functions* can be defined $\pi_i(x) = \Pr(F_i(x))$, and it may be approximated as $\pi_i(x) \approx 1$ when $\mathcal{K} \models F_i(x)$, $\pi_i(x) \approx 0$ when $\mathcal{K} \models \neg F_i(x)$ and $\pi_i(x) \approx (P_i^+ + P_i^-)/2$ otherwise. This induces the choice of weights: $w_i = 1/\widehat{\mathrm{var}}(F_i)$, for $i = 1, \ldots, m$ (to be possibly normalized).

These weights may be also employed to encode asymmetric *cost functions* (Duda et al., 2001; Hastie et al., 2001) to assign a different impact to the values that the dissimilarity on a given feature ($[\pi_i(a) - \pi_i(b)]^p$) may assume. These functions are based on a number of features that may be elicited from the knowledge base through suitable methods based on stochastic search (Fanizzi et al., 2008b).

Distance-based methods rely on a notion of similarity coded through a specific function for the target representations. We start off by presenting an adaptation of the *k-Nearest Neighbors* approach (d'Amato et al., 2008b) and then move to other learning schemes based on distances or basis function approximation (Fanizzi et al., 2009).

3.2 The Nearest Neighbors Approach

An instance-based procedure may exploit prototypical exemplars of the target class (examples) for predicting the membership of further individuals. In contrast to standard deductive instance-checking, such a method should also be able to provide an answer in cases where no answer can be inferred by deduction. Moreover, it may also provide a measure of the likelihood of its answer.

As in other density based methods, the idea is that similar instances (w.r.t. the one whose membership is to be determined) are likely to share the same membership locally. The objective is to induce an approximation for a discrete-valued target hypothesis function $h : IS \rightarrow V$ from a space of instances *IS* to a set of values V. In this setting, a predicted value indicates one in a set of pairwise disjoint classes. This case is dissimilar to the one of DL knowledge bases (as discussed in the previous section) where a resource can be instance of more than one concept at the same time. Furthermore, due to the OWA, there could be some cases for which it is not known if the resource belongs or does not belong to a given target concept. These cases should count as neutral (uncertain) information. Thus, an option can be to transform the multi-label prediction problems into ternary ones (d'Amato et al., 2008b), with $V = \{+1, -1, 0\}$. It is assumed that the hypothesis function values for the training instances are given by an expert, i.e. for the training instances we suppose that the

values of h_Q to be approximated may be determined as follows:

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

Let x_q be an instance whose class-membership is to be predicted. Using a similarity measure, the set of the k nearest training instances w.r.t. x_q is selected: $NN(x_q) = \{x_i \mid i = 1, \ldots, k\}$. The k-NN procedure approximates h on the grounds of the value that h is known to assume for the training instances in $NN_k(x_q)$. The value is decided by means of a weighted majority voting procedure.

Specifically, the estimate of the hypothesis function for the query individual is:

$$\hat{h}(x_q) = \underset{v \in V}{\text{argmax}} \left(\sum_{x_i \in NN_k(x_q)} \delta(v, h(x_i)) \cdot \sigma_q(x_i) \right) \tag{1}$$

where δ returns 1 in case of matching arguments and 0 otherwise, and σ_q is a function of the similarity w.r.t. the individual x_q whose membership is to be predicted. For instance, given a dissimilarity function d, $\sigma_q(x_i) = (d(x_i, x_q))^{-b}$ may be adopted (for some $b = 1, 2, \ldots$) (d'Amato et al., 2008b).

Note that, being based on a majority vote of the individuals in the neighborhood, this procedure should be robust in case of noise in the data (e.g. incorrect assertions), therefore, in contrast to a purely logic deductive procedure, it may predict the correct membership even in case of inconsistent knowledge bases, especially in densely populated regions of the instance space.

To force the procedure to provide a definite answer, a simplified setting may adopt the standard binary value set $V = \{-1, +1\}$, and the value of h_Q for a training instance x_i would be logically inferred or simply determined by the occurrence or absence of the corresponding assertion $Q(x_i)$ in the ABox (d'Amato et al., 2008b).

It should be noted that 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. In order to measure the likelihood of the decision made by the procedure (individual x_q belongs to the query concept denoted by value v maximizing the weighted votes in Eq. 1), given the nearest training individuals in $NN_k(x_q) = \{x_1, \ldots, x_k\}$, the quantity that determined the decision should be normalized (d'Amato et al., 2008b):

$$\ell(\hat{h}(\mathbf{x}_q) = v \mid NN_k(x_q)) = \frac{\sum_{i=1}^{k} w_i \cdot \delta(v, h_Q(x_i))}{\sum_{v' \in V} \sum_{i=1}^{k} w_i \cdot \delta(v', h_Q(x_i))} \tag{2}$$

Hence the likelihood of the assertion $Q(x_q)$ corresponds to the case when $v = +1$.

3.3 Reduced Coulomb Energy Networks

Further non-parametric approaches require the construction of inductive models obtained with a more complex training phase; such models can be implemented as the combination of approximating functions. An interesting example of such learning approaches is represented by the Reduced Coulomb Energy (RCE) network (Duda et al., 2001), which can be considered a simple form of the Radial Basis Function (RBF) networks and other artificial neural networks.

In the *training* phase a simple network based on prototypical individuals (parametrized for each prototype) is trained adjusting hypersphere radii around them w.r.t. their membership in some query concept (distinguishing membership/non-membership cases); This network is then exploited during the *classification* phase to make a decision on the class-membership of further individuals w.r.t. the query concept on the grounds of the membership related to the hyperspheres it lies in and the distance to the centers (prototypes).

The RCE Model In a categorical setting (Fanizzi et al., 2009), individuals shall be preliminarily projected onto $\mathbb{R}^m$ using suitable projection functions $\pi_i\colon\ \mathit{IS} \mapsto \mathbb{R}$ $(i = 1, \ldots, m)$.

The structure of a RCE network resembles that of a *probabilistic neural network* (Duda et al., 2001) with three layers of units linked by weighted connections, say w_{ij}'s and a_{jc}'s, for the input layer-to-hidden layer and hidden layer-to-output layer, respectively. The input layer receives its information from the individuals. The middle layer of patterns represents the features that are constructed for prediction; each node in this layer is endowed with a parameter ρ_j representing the radius of the hypersphere centered in the j-th prototype individual of the training set. During the training phase, each coefficient ρ_j is adjusted so that the spherical region is as large as possible, provided that no training individual of a different class is contained. Each pattern node is connected to one of the output nodes representing the predicted class. In this case two categories may be taken into account representing, resp., membership and non-membership w.r.t. the query concept Q.

Training Examples are made up of training instances labeled with their correct prediction $\langle \mathbf{x}_i, h_Q(\mathbf{x}_i)\rangle$, where $h_Q(\mathbf{x}_i) \in V$, as seen before. In this phase, each parameter ρ_j which represents the radius of a hypothetical m-dimensional hypersphere centered at the input example, is adjusted to be as large as possible (they are initialized with a maximum radius), provided that the resulting region does not enclose counterexamples. As new individuals are processed, each such radius ρ_j may be decreased accordingly (and can never increase). In this way, each pattern unit may correspond to a hyperspheres enclosing several prototypes, all having the same category label.

There are several subtleties that may be considered. For instance, when the radius of a pattern unit becomes too small (i.e., less than some threshold $\rho_{\min}$), this indicates highly overlapping different categories in a certain region. In that case, the pattern unit is called a *probabilistic* unit, and marked as such.

The method can be used both in *batch* and *on-line* mode. The latter incremental model is particularly appealing when the application may require performing intensive queries involving the same concept and new instances are likely to be made available over time. The method is related to other non-parametric approaches dealing with the instance density such as the one based on *Parzen windows* (Duda et al., 2001): however, the Parzen windows method uses fixed window sizes that could lead to some difficulties: in some regions a small window width is appropriate while elsewhere a large one would be better. The k-NN method uses variable window sizes increasing the size until enough samples are enclosed. This may lead to unnaturally large windows when sparsely populated regions are targeted. In the RCE method, the window sizes are adjusted until points of a different category are encountered.

Classification The prediction of the membership for a given query individual $\mathbf{x}_q$ using the trained RCE network is quite simple in principle. The set $N(\mathbf{x}_q) \subseteq \mathit{TrSet}$ of the nearest training instances is built on the grounds of the hyperspheres (determined by the ρ_j's) the query instance belongs to. Each hypersphere has a related prediction label determined by the prototype at its center. If all prototypes agree on the label this value is returned as the

induced estimate, otherwise the query individual is deemed as ambiguous w.r.t. Q, which represents the default case.

In case of uncertainty, this procedure may be enhanced in the spirit of k-NN classification recalled above. If a query individual is located in more than one hypersphere, instead of a catch-all decision requiring all involved prototypes to agree on the label to be assigned in a sort of *voting* procedure, each vote may be weighted by the similarity of the query individual w.r.t. the hypersphere center in terms of a similarity measure s, and the membership should be defined by the label whose corresponding prototypes are globally the closest, considering the difference between the closeness values of the query individual from the centers related to either label. Indeed, one may also consider the *signed* vote, where the sign is determined by the label of each selected training prototype, and sum up these votes determining the membership with a sign function.

Formally, suppose the nearest prototype set $N(\mathbf{x}_q)$ has been determined. The decision function is defined:

$$g(\mathbf{x}_q) = \sum_{\mathbf{x}_j \in N(\mathbf{x}_q)} h_Q(\mathbf{x}_j) \cdot s(\mathbf{x}_j, \mathbf{x}_q)$$

So, given a *tolerance* $\theta \in]0, 1]$ for the uncertain membership, if $|g(\mathbf{x}_q)| > \theta$ then the predicted value is $\mathrm{sgn}(g(\mathbf{x}_q))$ otherwise the membership remains undetermined. One may foresee that higher values of this threshold make prediction more *skeptical* in uncertain cases, while lower values make it more credulous in suggesting ± 1.

As previously noted, the analogical inference made by the procedure shown above is not guaranteed to be deductively valid. Indeed, inductive inference naturally yields a certain degree of uncertainty. In order to measure the likelihood of the decision made by the inductive procedure, one may resort to an approach that is similar to the one applied with the k-NN procedure. It is convenient to decompose the decision function $g(\mathbf{x})$ into three components corresponding to the values $v \in V : g_v(\mathbf{x})$ and use those weighted votes. Specifically, given the nearest training individuals in $N(\mathbf{x}_q) = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, the values of the decision function should be normalized as follows, producing a likelihood measure:

$$\ell(\hat{h}(\mathbf{x}_q) = v \mid N(\mathbf{x}_q)) = \frac{g_v(\mathbf{x}_q)}{\sum_{u \in V} g_u(\mathbf{x}_q)} = \frac{\sum_{j=1}^{k} \delta(v, h_Q(\mathbf{x}_j)) \cdot s(\mathbf{x}_q, \mathbf{x}_j)}{\sum_{u \in V} \sum_{h=1}^{k} \delta(u, h_Q(\mathbf{x}_h)) \cdot s(\mathbf{x}_q, \mathbf{x}_h)}$$

The likelihood of the assertion $Q(\mathbf{x}_q)$ corresponds to the case when $v = +1$ (i.e. to $g_1(\mathbf{x}_q)$). This could be used in case the application requires that the hits be ranked along with their likelihood values.

4 Learning with Kernels for the Semantic Web

Kernel methods are a field of research that has been widely studied over the last years. Kernel methods are particularly interesting from an engineering point of view because they allow for a separation of the learning algorithm and the choice of the data representation.

In the context of SW data, kernel machines can be applied for solving a variety of learning problems. The instances on which the kernels are defined are typically ABox elements. Kernel machines do not use the explicit representation of the training instances. Instead, they implicitly mimic the geometry of the feature space by means of a kernel function, a similarity function which maintains a geometric interpretation as the inner product of two vectors in some, potentially unknown, feature space.

While *Support Vector Machines (SVMs)* (Shawe-Taylor and Cristianini, 2004) for classification and regression are the best-known kernel machines, many other well-known learning algorithms can be "kernelized" as well. To make SW data accessible to these kernel machines thus requires the definition of suitable kernel functions.

Definition 1 (Kernel Function) Any function $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ on objects x, x' from some input domain $\mathcal{X}$ that satisfies $\kappa(x, x') = \langle \phi(x), \phi(x') \rangle$ is a valid kernel, whereby ϕ is a mapping function (feature representation) from $\mathcal{X}$ to some feature space $\mathcal{H}$.

For SW data, two different classes of kernel functions can be distinguished: those based on logical structures in the ontology and knowledge base (see Section 4.1) and those exploiting the graph structure of the SW data (see Section 4.2).

4.1 Kernel Functions based on Logical Structures

The kernels presented in this Section are based on the analysis of the logical properties of the instances. This means that the feature space consists of logical properties, e.g. type relations.

Gärtner et al. (2004) have proposed a principled framework for defining new kernels based on type construction, where types are defined in a declarative way. Given a set of type constructors, they propose to define one kernel per type constructor. The thus-defined kernels can then be combined using kernel modifiers such as sum and product.

Passerini et al. (2006) have proposed kernel functions on PROLOG proof trees. Here, individuals are described as first-order logic objects and the kernel function measures their similarity via the similarity of the proof trees of a special logic program.

The first kernel functions for SW data were restricted to the basic description logic $\mathcal{ALC}$ (Fanizzi and d'Amato, 2006, 2007). These kernel functions compare instances based on the structural similarity of the AND-OR trees corresponding to a normal form of the instances' concept descriptions (Baader et al., 2003). Their applicability is restricted due to the employment of the notion of (approximations of) *most specific concepts* (Baader et al., 2003) in order to lift instances to the concept-level where the kernels actually work. Additionally, the normal form of the concept descriptions is specific to the employed description logic. However, these kernels are not purely structural since they ultimately rely on the semantic similarity of the *primitive* concepts assessed by comparing their extensions (approximated by their retrieval) through a set kernel. Structural kernels for richer DL representations have been proposed by Fanizzi et al. (2008c). Here, the kernels from (Fanizzi and d'Amato, 2006, 2007) were extended to cover $\mathcal{ALCN}$. Further graph kernels for SW representations based on the idea of structural intersection/product are discussed in the next Sect. 4.2.

A definition of kernel functions for individuals in the context of the standard SW representations is reported by Bloehdorn and Sure (2007). The authors define a set of kernels for individuals based on their similarity with respect to the various kinds of assertions in the ABox (i.e. with respect to common concepts, datatype properties and object properties).

Bicer et al. (2011) have proposed kernel functions which are based on the grounding of specific ILP clauses in the RDF data graph. The actual kernel function is learned as a non-linear combination of simple clause kernels.

The kernels presented so far exploit specific language-dependent structures. A more flexible way of defining kernel functions is based on simple similarity functions parameterized on the semantics of instances w.r.t. a committee of concepts. Such kernels can be integrated with many efficient algorithms, that can implicitly embed feature selection. These

functions transform the initial representation of the instances into the related - so called - active features, thus allowing for learning the classifier directly from structured data (Cumby and Roth, 2003).

In this spirit, a different set of kernels, which is directly applicable to individuals, has been proposed by Fanizzi et al. (2008d). Given a dissimilarity measure d with values in $[0, 1]$ belonging to the family of pseudo-metrics defined by d'Amato et al. (2008b), the easiest way to derive a similarity measure would be: $\forall a, b\colon\ s(a, b) = 1 - d(a, b)$. For example, one may reuse the measure defined in Sect. 3.1, or define a family of kernels as follows. Given a pool of features $\mathsf{F} = \{F_1, F_2, \ldots, F_m\}$, the family of *kernel functions* is defined:

$$k_p^{\mathsf{F}}(a, b) = \left[\sum_{i=1}^{m} |w_i \kappa_i(a, b)|^p\right]^{1/p}$$

where $p > 0$, w_i are weights reflecting the importance of a certain feature in discerning the various instances (d'Amato et al., 2008b) and $\forall i \in \{1, \ldots, m\}$ the *simple kernel function* for F_i is defined:

$$\kappa_i(a, b) = \begin{cases} 1 & (\mathcal{K} \models F_i(a) \wedge \mathcal{K} \models F_i(b)) \vee (\mathcal{K} \models \neg F_i(a) \wedge \mathcal{K} \models \neg F_i(b)) \\ 0 & (\mathcal{K} \models \neg F_i(a) \wedge \mathcal{K} \models F_i(b)) \vee (\mathcal{K} \models F_i(a) \wedge \mathcal{K} \models \neg F_i(b)) \\ s_i & otherwise \end{cases}$$

The rationale for these kernels is that the similarity between individuals is determined by their similarity w.r.t. each concept in a given committee of features (a sort of similarity *context*). Two individuals are maximally similar w.r.t. a given concept F_i if they exhibit the same behavior, i.e. both are instances of the concept or of its negation. Because of the *open-world* semantics, a reasoner may be unable to ascertain the concept-membership of some individuals, hence, since both possibilities are open, an intermediate value is assigned to reflect such uncertainty. Experiments regarding instance-based classification demonstrated the effectiveness of the kernel using the very set of both primitive and defined concepts found in the knowledge bases. However, the choice of the concepts to be included in F is crucial and may be the object of a preliminary learning problem to be solved through stochastic search, as in (Fanizzi et al., 2008b).

The kernels defined by Bloehdorn and Sure (2007) are contained in the family of kernel functions defined by Fanizzi et al. (2008d). This is because each of those kernel functions is based on a specific set of simple common features and thus can be simulated given proper class expressions. For instance, the *common class* kernel defined by Bloehdorn and Sure (2007) is equivalent to the kernel function presented above when only named concepts are considered as features. In this perspective the family of kernels represents a generalization allowing for more complex features (and dimensions).

4.2 Kernel Functions based on Graph Structures

SW data can be represented in the form of directed labeled graphs. It is thus possible to apply kernels developed for general graphs for learning from semantic data. In particular, graph kernels have been used for classifying entities in the data set. After a suitable representation of the entity is obtained (usually in form of a graph representing the entity's neighborhood) which we call *instance graph*, graph kernels can be used for comparing the entities by comparing the corresponding instance graphs. The use of graph kernels in this setting implies that the semantics of the represented data is not exploited explicitly, but through analyzing

the structure of the corresponding graphs. The advantage of this approach is that no assumptions on the structure of the data have to be made beforehand and that no manual definition of features is necessary.

A wide range of graph kernels have been proposed in the literature. All these functions rely on counting subgraphs of a specific structure in the graphs. While the subgraph isomorphism problem, i.e. the problem of deciding whether a graph contains another graph as subgraph, is known to be NP-complete, the search for common subgraphs with specific properties can be performed more efficiently.

Horvth et al. (2004) have proposed a kernel which is based on counting common cyclic and tree-like patterns in undirected labeled graphs, independent of their frequency.

Shervashidze and Borgwardt (2009) defined a kernel which is based on counting common subtree-patterns of the input graphs. Only complete tree matches are counted.

Gärtner et al. (2003) have defined a kernel which is based on counting walks up to a certain length in the graph. The feature space consists of one feature for each possible walk in the graph, the feature weight represents how often this walk has been found in the graph. The kernel function can be calculated efficiently through the calculation of a matrix power series based on the adjacency matrix of the two input graphs.

All these kernels share two shortcomings with respect to their application on SW data: they were devised for relatively small graphs and work more efficiently on graphs with few distinct node labels. However, on the SW labels, i.e. URIs, are used to identify nodes and neighborhood graphs grow exponentially with a growing number of hops being included in the neighborhood, making instance graphs large: for an instance definition using a two hop neighborhood of the entity of interest, the graphs obtained in specific datasets are twice as big as the biggest graphs used in classic graph kernel benchmarks.

Most recently we have been developing graph kernels that work efficiently on graphs exhibiting the specific properties of RDF graphs. The calculation of the kernel function is based on the *intersection graph*:

Definition 2 (Intersection Graph) The intersection graph $G_1 \cap G_2$ of two graphs G_1 and G_2 is defined as: $V(G_1 \cap G_2) = V_1 \cap V_2$ and $E(G_1 \cap G_2) = \{(v_1, p, v_2) | (v_1, p, v_2) \in E_1 \wedge (v_1, p, v_2) \in E_2\}$.

The intersection graph contains all elements that occur in both input graphs. It is equivalent to the product graph used by Gärtner et al. (2003) in the case of RDF data. Kernel values are obtained by counting specific structures - like relations, walks or paths - in the product graph.

A more efficient way of calculating the common elements of the two input graphs is proposed by means of the *intersection tree*, a structure similar to the intersection graph, which can directly be extracted from the data graph and thus does not require the computation of the two instance graphs.

5 Multivariate Prediction Models for the Semantic Web

The basis of all representations proposed for the SW, e.g. RDF, is a labeled graph that provides a natural representation for entities and their mutual relationships, although it should be emphasized that just a small subset of the more expressive DL can be represented as graphs. In machine learning, multi-relational graphs are addressed in Statistical Relational Learning (SRL) (a recent overview on Statistical Relational Learning can be

found in (Getoor and Taskar, 2007)). This section and the following two sections (Sec. 6 and Sec. 7) provide an overview on SRL approaches that have been applied in the SW context. The goal in all three approaches is to predict the probabilities of RDF-triples being true based on implicit patterns found in the data. In this section, we discuss scalable approaches that are based on the decomposition of matrices and tensors derived from the RDF-graph. In Sec. 6 we describe how probabilistic graphical models have been derived from the RDF-graph. There, random variables in the probabilistic graphical model represent the truth values of RDF-triples. Finally, in Sec. 7, we discuss SRL approaches that exploit first-order logical constructs, incorporating logic-based background knowledge in statistical modeling.

5.1 Semantic Web Learning based on a Matrix and Tensor Factorization

Our discussion follows the discussion in (Tresp et al., 2008), (Yu et al., 2005), and (Yu et al., 2006). Consider entities of a certain type (e.g., person) and the associated `(subject, predicate, object)` triples with these entities being the subjects. These entities (i.e., the subjects) form the rows in the data matrix and the `(predicate, object)` pairs define the columns. A matrix entry is equal to one if the corresponding triple is present in the data and is zero otherwise. One considers `(predicate, object)` pairs with a minimum frequency in the data, thus limiting the number of columns. Additional columns can be defined by aggregated information (e.g., the average income of friends can be valuable information if it is the goal to predict a person's income.). The ideas is now to apply matrix factorization which yields an approximation to the original matrix. After matrix reconstruction, the now non-zero matrix elements can be interpreted as confidence values that the corresponding RDF-triples should be true. This approach is motivated by the highly-scalable matrix factorization developed for recommendation systems (see the Netflix challenge [3] and (Koren, 2008)). The major difference is that recommendation systems only consider a single relation-type, whereas in the SW domain, one works with multiple relation-types. Mathematically, one achieves a multivariate prediction task, where the RDF-triples form the multivariate targets and the aggregated information forms the inputs.

By using aggregation, the approach suffers the same limitations as propositionalization if applied to complex SW ontologies. Thus, in contrast to the approaches presented in Sec. 6 and 7, the presented approach is not able to encode recursive dependencies.

Any suitable and scalable matrix factorization approach can be used. Popular examples are matrix completion methods based on an eigenvector analysis of the data matrix (e. g. , *Singular Value Decomposition* (SVD)), e. g. , (Lippert et al., 2008), matrix completion based on Non-Negative Matrix Factorization (NNMF) as in (Lee and Seung, 1999) and matrix completion using *Latent Dirichlet Allocation* (LDA) (see (Blei et al., 2003)). NNMF is a decomposition under the constraints that all terms in the factoring matrices are non-negative. If the data also contains information about triples that are known to be false, the scalable approaches used in the Netflix competition might be used (see (Takacs et al., 2007)).

The presented approach has been implemented in the EU FP7 LarKC project as the SUNS model (see (Huang et al., 2009)). SUNS has successfully been applied to predicting friendship patterns in social networks, to predict gene-disease relationships in the life-sciences, to predict party membership in DBpedia data and to recommends services in E-commerce applications. In the latter, services useful for a certain mashup application were

[3] `http://www.netflixprize.com/`

predicted based on services already being used exploiting service patterns in other applications. The SUNS approach was solving the personalization tasks in the winning entry of the ESWC 2011 AI Mashup challenge [4] and the ISWC 2011 Semantic Web competition [5].

Tensors generalize matrices to three and more modes and permit interesting generalizations to the basic SUNS approach. In particular, tensor models permit the inclusion of contextual information. Let's consider two of the leading approaches. The first one concerns the Pairwise Interaction Tensor Factorization (Rendle and Schmidt-Thieme, 2010; Rendle et al., 2010; Karatzoglou et al., 2010) and its generalizations. A combination of Markov chains with matrix factorization was applied to next-basket recommendation in eCommerce applications. In their work the authors introduce a generic way to handle the temporal information by factorizing a transition cube. As an example, a person might have a general preference for science fiction movie and might be more likely to view Star Trek or Star Wars series in their natural order. The approach proposed by Wermser et al. (2011) is an extension to (Rendle et al., 2010) and generalizes it to a larger class of recommendation scenarios which can model any context like spatial and temporal data. This kind of rich spatial and temporal information is becoming increasingly available on the Semantic Web and can be leveraged to improve predictive performance. A probabilistic and modular variant based on the idea of decomposable graphical models has been described in (Tresp et al., 2011).

In the second type of approach, one exploits the fact that the `(subject, predicate, object)` structure maps nicely onto a 3-way matrix where the subject is represented by the first mode, the predicate by the second mode and the object by the third mode (Nickel et al., 2011). Thus a complete RDF-store can be mapped onto a tensor and can be analyzed by the appropriate tensor factorization approaches and, thus, the different relation-types can share statistical strength. The particular tensor model developed in (Nickel et al., 2011) is able to perform collective learning and thus depends less on the explicit aggregation of information.

In summary, due to their scalability and essentially also their simplicity, matrix and tensor factorization are very promising machine learning approaches for modeling the multi-relational dependencies in the semantic web. In.particular their strength is in the exploitation of statistical patterns that are far from deterministic. By exploiting sparsity, both matrix an tensor factorization has been applied to millions of entities where the factorized model can estimate the truth values of 10^14 statements. We describe one concrete application of multivariate methods to SW data in Sec. 8.2.1.

6 Relational Graphical Models for the Semantic Web

The approaches described in Sec. 3 and Sec. 4 aim at comparing instances of relational data. In contrast, the matrix decomposition approaches in Sec. 5 predict the truth values of all possible statements of usually one or a small number of relations in the SW. Unlike the matrix decomposition techniques and the distance based techniques, Relational Graphical Models (RGM) are probabilistic models where statements are represented by random variables and a whole domain is modeled in one coherent framework.

Graphical models model dependencies in structured domains, like the SW, and obtain efficiency by exploiting probabilistic independencies and have successfully been applied in a number of machine learning approaches. In graphical models, nodes correspond to random variables, like persons in the social network example, and links between nodes, like

[4] `https://sites.google.com/a/fh-hannover.de/aimashup11/`

[5] `http://www.cs.vu.nl/~pmika/swc/challenge.html`

the knows relation between persons, indicate probabilistic interactions. The most important examples of graphical models are Bayesian networks and Markov networks. Bayesian networks employ directed links, that often have a causal interpretation. Heckerman et al. (1995) provides an excellent introduction to learning in Bayesian networks. Markov networks employ undirected links and are the basis, e.g., for conditional random fields (Lafferty et al., 2001). Graphical models can contain latent variables whose states are unknown but which might be estimated from the data. Prominent examples here are probabilistic mixture models.

In a standard approach, there is an instance of a graphical model for each statistical unit (e.g., for each patient) and the random variables describe attributes (e.g., patient attributes such as age, gender, symptoms, diagnosis). Here, given parameters, the probability distribution factors over data points.

Relational graphical models have been developed in the context of multi-relational data models, plate models and entity-relationship models (Getoor and Taskar, 2007), and can be easily adapted to domains such as the SW, where relationships play a prominent role. Modeling becomes more complex, since relationships become random variables that might have impact on attributes. As an example, the wealth of friends might be correlated. Thus if one wants to predict friendship, one needs to consider the wealth of the involved persons and, vice versa, if one wants to predict the wealth of a person, one needs to consider the wealth of this person's friends. In the most straightforward encoding of SW data, RDF-triples become binary random variables in the relational graphical model. If a random variable assumes the value one, then the corresponding triple is true and if a random variable assumes the value zero, then the triple is false. Values in between indicate certainty values of a triple.

Due to this inter-connectivity, probability distributions do not factor into data points: in essence, a whole data set must be treated as one data point, which makes inference and learning more complex. However, there as been efforts in scaling graphical models to large data sizes by efficient inference techniques (Porteous et al., 2008) or distributed processing (Newman et al., 2007).

In the following subsection we will briefly discuss three examples of graphical relational model, i.e., probabilistic relational models, relational Markov networks and latent-class relational graphical models. In Sec. 7 these models are applied in the context of logical constraints as they occur in the SW.

6.1 Probabilistic Relational Models (PRMs)

Probabilistic relational models were one of the first published directed RGMs and found great interest in the statistical machine learning community (see (Koller and Pfeffer, 1998; Getoor et al., 2007)). PRMs combine a frame-based logical representation with probabilistic semantics based on directed graphical models. The nodes in a PRM model constitute the probability distribution of object attributes whereas the relationships between objects are assumed known. Naturally, this assumption simplifies the model greatly. PRMs have been extended to also consider the case that relationships between objects are unknown, which is called *structural uncertainty* in the PRM framework (see (Getoor et al., 2007)). The simpler case, where one of the objects in a statement is known, but the partner object is unknown, is referred to as *reference uncertainty*. In reference uncertainty the number of potentially true statements is assumed known, which means that only as many random nodes need to be introduced. The second form of structural uncertainty is referred to as *existence uncertainty*,

where binary random variables are introduced representing the truth values of relationships between objects.

For some PRMs, regularities in the PRM structure can be exploited (encapsulation) and exact inference is possible. Large PRMs require approximate inference; commonly, loopy belief propagation is being used. Learning in PRMs is likelihood-based or based on empirical Bayesian learning. Structural learning typically uses a greedy search strategy, where one needs to guarantee that the ground Bayesian network does not contain directed loops.

6.2 Relational Markov Networks (RMNs)

Relational Markov networks generalize many concepts of PRMs to undirected RGMs (Taskar et al., 2002). A RMN specifies a conditional distribution over all of the labels of all of the entities in an instantiation given the relational structure.

Intuitively speaking, it specifies the cliques and potentials between features of related entities at a template level, so a single model provides a coherent distribution for any collection of instances from the schema. To specify what cliques should be constructed in an instantiation, RMNs use conjunctive database queries as clique templates. By default, RMNs define a feature function for each possible state of a clique, making them exponential in clique size.

RMNs are mostly trained discriminately, as they define a conditional distribution with a set of random variables to condition on and a set of target variables.

In Sec. 7 we describe Markov logic networks, which are closely relate to RMNs and which exploit logical constructs, as they are present in the SW.

6.3 Latent-Class Relational Graphical Models

A latent-class RGM incorporates latent or hidden variables which can be interpreted as cluster variables. The *infinite Hidden Relational Model* (IHRM) (Xu et al., 2007) is an example of a directed latent-class RGM. In an IHRM a latent variable is introduced for each entity and this variable is the parent of all variables that this entity is involved in. These are the only links in the relational graphical model. As an example, the latent variable of Jack is the parent of the node representing the triple (Jack, income, High) and is also the parent of the node representing the triple (Jack, friendOf, Jane).

In the IHRM, the number of states in each latent variable may potentially be infinite by using the formalism of *Dirichlet process mixture models*. However, in inference, only a small number of the infinite states are occupied, leading to a clustering solution where the number of states in the latent variables is automatically determined during inference.

Since the dependency structure in the ground Bayesian network is local, one might get the impression that only local information influences prediction. In fact this is not the case, since in the ground Bayesian network common children with evidence lead to interactions between the parent latent variables. Thus, information can propagate in the network of latent variables. Training is usually based on forms of Gibbs sampling.

The IHRM has a number of key advantages, when applied to the SW. First, no structural learning is required, since the directed arcs in the ground Bayesian network are directly given by the structure of the SW graph. Second, the IHRM model can be thought of as an infinite relational mixture model, realizing hierarchical Bayesian modeling. Third, the mixture model allows a cluster analysis providing insight into the relational domain.

In Sec.7, we show how ontological constraints can be incorporated in training an IHRM and in Sec. 8.2.2 we describe experimental results.

7 Towards First-order Probabilistic Learning for the Semantic Web

The approaches described so far do not rely on logical representations to construct models or define semantics and inference capabilities. At the most, they use formalisms like ER-models as a concise way to describe atomic ground networks of facts. However, as described in Sec. 2, SW ontologies were designed for the use of formal knowledge representation languages like description logic. Thus, in order to apply machine learning in formal ontologies, while utilizing the deductive reasoning capabilities and expressive power of the logical representation, the inference methods ultimately should be able to incorporate both, logical deductive inference and probabilistic inductive inference.

In the past years an increasing number of different SRL techniques with expressive formal representations have been proposed. Two different approaches can be identified. First, models that start with an existing logical theory and extend it with probabilities and second, statistical models that are extended to structured data formats. Sec. 3-6 concentrated on the second type and in Sec. 2.3 some formalisms of the first type were mentioned. In the remainder of this section, we will touch on research that strives for formalisms that provide a high expressive logical representation and full deductive and inductive inference capabilities. We will call approaches from this research area *first-order probabilistic learning*.

First-order logic (FOL) is in general assumed to be more general than DL, thus, first-order probabilistic inference is equipped to use the full potential of formal SW languages in a deductive and inductive way. First-order probabilistic logics also have the potential for more powerful and efficient inferencing, like the use of quantifiers for handling infinite numbers of possible worlds.

Here, we review fundamental research on approaches to probabilistic FOL inference which has the potential to contribute to the SW, once expressive ontologies are available. Those approaches have not yet been applied to the SW as there are (i) no suitable data sets (in terms of expressivity and number of instances) on the SW yet and (ii) the listed techniques suffer from scalability problems.

Thus, this section might not be of interest in terms of what can already be achieved on the SW today, as it was the focus of the sections before. However, we think the formalisms outlined here provide a vision of what needs to be achieved if a SW with more general and powerful knowledge representation and reasoning capabilities should become a reality in the more distant future.

7.1 First-Order Probabilistic Representations, Inference and Learning

All first-order probabilistic learning approaches can be characterized in terms of three aspects: First, their representation of logical formulae and probabilities, second their inference mechanisms and third, the learning or inductive inference mechanisms. Those three topics will be selectively covered on a high-level in the following subsection.

7.1.1 First-Order Probabilistic Representations

As shown in Sec. 2 the types of constructs that might be used to express formal knowledge in the SW can be defined in a formal language like description logic (DL). As most DLs

are decidable fragments of first order logic, first-order probabilistic representations - in theory - inherently provide the expressivity needed for formal ontologies. Thus, we can cover first-order probabilistic representations in a general manner, without being concerned about the details of concrete ontology languages and their constructs like hierarchies, cardinality restrictions, role reflexivity, role disjointness, and so on.

Possible World Models: Most first-order probabilistic models make use of the notion of *possible worlds* to provide semantics for probabilistic statements. Statements in one such instantiation (possible world) can either be true or false. This imposes a restriction on the states of each possibly statement (ground atom) depending on the elements known to be true or false in this instantiation.

Next this ground atom is assigned a random variable stating the probability that this ground atom is true over all possible combinations of ground atoms in all possible worlds. The estimation of this probability hence is a marginalization since it is computed by summing over possible world probabilities.

An assignment of truth values to all random variables is called Herbrand interpretation. The set of all possible worlds is called Herbrand base. The approaches using possible world semantics differ in how these probabilities are defined and mapped to random variables, and how they are learned and used for inference.

The next step needed to get a full probabilistic model is to combine the probabilities on the truth values of atomic formulae. This defines a joint distribution over truth values of atomic formulae. If each possible world has a conjunction that it alone satisfies, this will result in a complete distribution over possible worlds. An interpretation for a relation is a set of true ground atomic formulae with the relation as the predicate symbol. This can vary across possible worlds.

In conclusion, those possible world models can incorporate both, the probabilities of certain constructs from observed instance data and expressive constructs as provided by e.g., DL axioms.

Entailment and Proof based Models Possible world models can be seen as a means to upgrade graphical models to a relational representation (for more on upgrading see (De Raedt, 2008)). Learning approaches that avoid the construction of Herbrand interpretations by not explicitly encoding a set of joint distributions over possible worlds are learning by entailment and learning from proofs. Both approaches extend techniques from *Inductive Logic Programming* (ILP) to probabilistic settings.

ILP models use *definite clauses* as key concepts to represent hypotheses. A definite clause cl is a formula of the form

$h^1 \models h^2, h^3, ...$

where h are logical atoms and $\models$ is the entailment relation. A clause cl entails another clause cl' if each model of cl is also a model of cl'. A logic program consists of a set of clauses and its semantics is defined by all facts (ground atoms) which can be logically entailed. The process of deducing facts is called resolution. ILP is concerned with finding a hypothesis in form of a logic program from a set of positive and negative examples.

Such representation comply with rule languages for the SW like the Semantic Web Rule Language (SWRL) (Horrocks et al., 2004) or the Rule Interchange Format (RIF) (Kifer, 2008) and can be extended to probabilistic rules: In Stochastic ILP (SILP) settings, probabilities (more precisely weights) are not attached to facts, but to the entailment relation in definite clauses or to resolution steps in a proof. Thus, the two directions are often called *learning from entailment* and *learning from proofs* . There are also some approaches which build

upon other types of 'uncertain logic', for example (Carbonetto et al., 2005). For more details on probabilistic ILP approaches to first-order probabilistic learning see e. g. , (De Raedt et al., 2008).

7.1.2 First Order Probabilistic Inference

One advantage of first-order probabilistic models compared to non logic-based formalism is that they are potentially more powerful in inferring additional knowledge that is not explicitly given in the data. We will mention some areas where this potential is being used.

Knowledge Based Model Construction Constructing a ground network of probabilities of atomic formulae is not feasible in many real world scenarios. Thus, formalisms were introduced that provide mechanisms for defining probabilities which must be inferred rather than just 'looked up'.

Many methods use deductive inference to construct only the relevant parts of those models. This procedure is known as *knowledge based model construction* (De Raedt et al., 2008). For instance, if the query to be answered is known in advance only the relevant ground atoms need to be considered to answer the query. The same applies to discriminative setting where only one target value is of interest. Again, the full ground network is never actually constructed. Instead, only just enough of it is constructed to answer the given probabilistic query.

Constraining by Satisfiability The fundamental inference task in DL is checking the satisfiability of the knowledge base. This is traditionally utilized for standard deductive inference tasks like instance membership checking or subsumption. The Infinite Hidden Semantic Model (IHSM) proposed by Rettinger et al. (2009) combines the latent-class relational learning of the IHRM (cmp. Sec. 6.3) with a constraining of possible predictions using DL satisfiability checking. In this way, hard constraints can inherently be enforced during prediction.

The setup is comparable to "learning over constrained output" by [Punyakanok et al., 2005] where the setup relates to structured output classification (see Sec. 5). Additionally "learning over constrained output" considers complex dependencies among the output variables which are captured by prespecified hard constraints. Using logical deduction, the logical theory can be used to check the satisfiability of arbitrarily complex constraints. This is a clear advantage over those of linearly separable classifiers as in [Punyakanok et al., 2005].

We describe one application of the IHSM in the experimental section (Sec. 8.2.2).

Lifted Probabilistic Inference As mentioned before, the most powerful advantage of first-order probabilistic models is that they have the potential to make use of (probabilistic) first-order logical inference[6]. So far, the approaches introduced need to generate all related instances to calculate the probability of one unknown fact. This sharply contrast to inference procedures in non-probabilistic first-order logic or clausal logic like those based on resolution. In resolution grounding is avoided whenever possible, which can make inference much more computational efficient. Thus, first-order logical inference can deal with a potentially infinite universe as it is necessary when allowing quantifiers, partially observable data and an open world assumption.

[6] This relates to *deduction* in non-probabilistic logic

Even at the rudimentary level of defining probabilities for atomic formulae some of the power of first-order methods is apparent. The basic point is that by using variables one can define probabilities for whole families of related atomic formulae. For instance, sometimes it is sufficient to calculate a probability when the number of true related ground instances is known. Which can be done without actually constructing all inferences and counting them. This type of inference is called *lifted probabilistic inference* and is largely an open research question (Poole, 2003; Milch et al., 2008; De Salvo Braz et al., 2005).

7.1.3 First Order Probabilistic Learning

Lifted probabilistic inference is concerned with inferring the probabilities of unknown facts given the first-order probabilistic model with known parameters. The correlated inductive inference task is how the model and its parameter are learned from observations. This section does not go into technical details of specific approaches because there are numerous different ways of learning models and parameters depending on the concrete method.

As mentioned before, there are two main elements that need to be learned to achieve a probabilistic model that can then be used for inference. On the one hand, the structure (or the logical program) needs to be learned if it is not pre-specified by hand. Commonly, a search through the space of possible structures is performed. In most of the cases, some formulae are defined by a domain expert *a priori*. Additional formulae can then be learned by directly optimizing the pseudo-likelihood cost function or by using ILP algorithms. In the first-order learning case the sets of random variables in each of the example interpretations should correspond to those of the probabilistic logic.

On the other hand if the model is given (or learned), parameters need to be estimated. If a grounded GM can be constructed standard parameter estimation techniques for Bayesian and Markov networks can be applied. For the lifted case corresponding parameters of the different instantiations are 'tied together' by the clauses defined by the first-order probabilistic model.

7.2 First-Order Probabilistic Methods and Applications

So far we gave an abstract overview of general ideas and concepts used in first-order probabilistic methods. This section provides a selective overview of concrete probabilistic methods which incorporate both, logic and probability and show the potential for the application of first-order probabilistic inference to SW data.

As mentioned in the previous section, logic based probabilistic formalisms usually define probabilities in two ways. On the one hand, probabilities over interpretations are used which is mostly done using graphical models. Popular formalisms of this type are for example, Bayesian Logic Programs (Kersting and De Raedt, 2001) (BLP) and relational Bayesian networks (Jaeger, 1997). A popular undirected approach of this type is Markov Logic Networks (MLN). BLP and MLP as the most popular representatives will be introduced briefly in the next section.

On the other hand there are approaches using probabilities over proofs or entailments like probabilistic logic programming (Ng and Subrahmanian, 1990), independent choice logic (Poole, 1997), stochastic logic programs (Muggleton, 1996) and PRISM (Sato et al., 2005).

7.2.1 Bayesian Logic Programs

A *Bayesian logic program* (BLP) (Kersting and De Raedt, 2001) is an intuitive extension of previously described logic programs to a Bayesian setting. BLPs are defined as a set of definite and Bayesian clauses. A Bayesian clause specifies the conditional probability distribution of a random variable given its parents on a template level, i.e. in a node-class.

$$R^1|R^2, R^3, ...$$

Note, the similarity to a definite clause in ILP which naturally maps to SW rule languages. In a BLPs, for each clause there is one conditional probability distribution and for each Bayesian predicate (i.e., node-class) there is one combination rule.

A special feature of BLPs is that, for a given random variable, *several* such conditional probability distributions might be given. As an example, $knows(e_1, e_2)|knows(e_1, e_3), knows(e_3, e_2)$ and $knows(e_1, e_2)|knows(e_2, e_1)$ specify the probability distribution for a $person_1$ knowing another $person_2$ if $person_2$ is known by a friend of $person_1$ or if $person_1$ is known by $person_2$. In this case the the truth value for
$knows(e_1, e_2)|knows(e_1, e_3), knows(e_3, e_2), knows(e_2, e_1)$ can then be calculated based on various combination rules (e. g. , noisy-or).

BLPs use knowledge-based model construction to ground the Bayesian clauses. The result is a propositional Bayesian Network which defines the joint probability distribution. Then standard Bayesian network learning and inference like the EM-algorithms are used to learn parameters and predict probabilities of facts.

Relational Bayesian networks (Jaeger, 1997) are related to Bayesian logic programs and use probability formulae for specifying conditional probabilities.

7.2.2 Markov Logic Networks

Markov Logic Networks (MLN) combine first-order logic with Markov networks and thus can incorporate DL formulae. However, the logical formulae are seen as soft constraints on the set of possible worlds. If an interpretation does not satisfy a logical formula it becomes less probable, but not impossible as in IHSM. In a MLN this is realized by associating a weight with each formula. The larger the weight, the higher is the confidence that a formula is true. When all weights are equal and become infinite, one strictly enforces the formulas and all worlds that agree with the formulas have the same probability.

Let F_i be a formula of first-order and let $w_i \in \mathbb{R}$ be a weight attached to each formula. Then a MLN L is defined as a set of pairs (F_i, w_i) (Richardson and Domingos, 2006; Domingos and Richardson, 2007), where F_i is a formula in first-order logic and w_i is a real number. One introduces a binary node for each possible grounding of each predicate appearing in L given a set of constants $e_1, \ldots, e_N$, where N is the number of entity instances. The state of the node is equal to 1 if the ground atom/statement is true, and 0 otherwise (for an s-ary predicate there are N^s such nodes).

Thus, the nodes in the Markov network are the grounded predicates. In addition the MLN contains one feature (cmp. to Markov networks) for each possible grounding of each formula F_i in L. The value of this feature is 1 if the ground formula is true, and 0 otherwise andw_i is the weight associated with F_i in L. A Markov network $M_{L,\{e\}}$ is a grounded Markov logic network of L with

$$P(\mathbf{U} = \mathbf{u}) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(\mathbf{u})\right)$$

where $n_i(\mathbf{u})$ is the number of formula groundings that are true for F_i. MLN makes the unique names assumption, the domain closure assumption and the known function assumption.

Like BLPs, MLNs first construct all groundings of the first-order formulae. Then Markov networks inference can be used. The simplest form of inference concerns the prediction of the truth value of a grounded predicate given the truth values of other grounded predicates. In the first phase, the minimal subset of the ground Markov network is constructed by knowledge-based model construction that is required to calculate the conditional probability. In the second phase, usually Markov Chain Monte Carlo algorithms are used to approximate the truth values in this reduced network.

Thus, learning consists of estimating the weights w_i. In learning, MLN makes a closed-world assumption and employs a pseudo-likelihood cost function, which is the product of the probabilities of each node given its Markov blanket. The basic Gibbs step consists of sampling one ground atom given its Markov Blanket. The Markov blanket of a ground atom is the set of ground predicates that appear in some grounding of a formula that the atom contains.

8 Selected Experimental Results on Semantic Web Mining Tasks

As motivated in Sec. 1, data mining methods have the potential to solve many pressing tasks resulting from the increasing amount of linked RDF data on the web as introduced in Sec. 2.1. In addition more sophisticated ML methods like the ones described in Sec. 7 are well suited for more expressive logical representation as they are expected to be used in the future internet and introduced in Sec. 2.2.

Although SW knowledge bases have become widely available (Bizer et al., 2009), applications of ML algorithms to those data representations are still rare. This is partly due to the special demands to learning algorithms in this complex relational domain, which we discussed in the previous sections. More importantly, the dilemma of real world formal ontologies is that they either are expressive or large scale, but not both. This means that they either provide a very elaborate formalization of the underlying structure like a schema or *TBox* but do not provide instance knowledge and observations in large quantities at the same time or vice versa.

This section is intended to give an overview of real world SW-related data mining tasks as listed in Sec. 2 by reporting our recent experimental results. The algorithms applied are all discussed in the previous sections (Sec. 3 - Sec. 6) Most of the results have been published before (cmp. Huang et al. (2010); Rettinger et al. (2009)), but we added some new results or summarized existing results to draw concise conclusions.

We start with experiments concerning the most basic ML task, namely class-membership prediction (Sec. 8.1.1), i.e. the assignment of individuals to classes, like person-instances to occupation-classes (also known as instance-checking in DL reasoning). In the perspective of Semantic Web applications backed by expressive ontologies, it is shown that this crucial task can be successfully tackled by resorting to ML methods and resulting inductive models. Another classification task is the prediction of features of instances, like the age or gender of a person, which we will call *property value prediction* as termed in RDF and present experiments in Sec. 8.1.2.

A machine learning task that has been of increasing interest mainly for recommendation engines and has been the subject of research in statistical relational learning is *relation* or *link prediction*. Typical examples are movie, weblink or friend recommendations to persons

(see Sec. 8.2). First, we report results on basic link prediction where we also investigate the question of how different data crawling strategies influence the performance (see Sec. 8.2.1). Then, we investigate how incorporating more expressive ontological constructs can affect the outcome (see Sec. 8.2.2).

Another important conventional data mining task is the *clustering* of instances like similar persons which we present in Sec. 8.3.

8.1 Classification

8.1.1 Class-Membership Prediction with Expressive Ontologies

A first experiment was designed in order to evaluate the effectiveness of classification models learned with inductive methods (surveyed in Sects. 3-4) compared to the standard purely logic procedures on a series of class-assignment problems. The knowledge bases employed in the experiment, which may be considered as a sort of logically encoded *background knowledge* (DL axioms), were selected among the many OWL ontologies designed to models specific domains for Semantic Web applications. Approximate class-membership prediction using inductive procedures can not only provide correct results but also propose non-logically derivable predictions, by exploiting the implicit information contained in the data (*ABox* assertions).

Data Set: A number of OWL ontologies modeling various domains were selected (most of them employed in the mentioned and related experiments), namely: WINE, SURFACE-WATER-MODEL (SWM), and NEW TESTAMENT NAMES (NTN) from the Protégé library[7], the FINANCIAL ontology[8] used in the *SEMINTEC project*, the *BioPax* glycolysis ontology[9] (BIOPAX), the *Semantic Web Service Discovery* dataset[10] (SWSD), one of the ontologies randomly generated by the *Lehigh University Benchmark*[11] (LUBM) and an exemplar of a business process repository obtained exploiting the *Business Process Modelling Notation* ontology[12] (BPMN).

Artificial problems were created as follows. For each ontology, 30 target concepts were randomly generated using concepts and roles defined therein. The examples were labeled ($V = \{-1, 0, +1\}$) according to the reasoner response (assuming the role of an expert) w.r.t. each target concept. For each concept two classification procedures are employed. One exploits the implementation of a SVM from the LIBSVM library[13] (default settings were used) coupled with a kernel function for individuals in DL ontologies (see Sect.4). The other is based on the k-NN procedure described in Sect. 3.2 (with $k = \sqrt{N}$, where N was the number of training instances considered).

[7] http://protege.stanford.edu/plugins/owl/owl-library

[8] http://www.cs.put.poznan.pl/alawrynowicz/financial.owl

[9] http://www.biopax.org/Downloads/Level1v1.4/biopax-example-ecocyc-glycolysis.owl

[10] https://www.uni-koblenz.de/FB4/Institutes/IFI/AGStaab/Projects/xmedia/dl-tree.htm

[11] http://swat.cse.lehigh.edu/projects/lubm

[12] https://dkm.fbk.eu/index.php/BPMN_Ontology

[13] http://www.csie.ntu.edu.tw/~cjlin/libsvm

Table 1 Results of the experiments on instance-checking using a model learned by a soft-margin SVM.

ontology	match	commission	omission	induction
SWM	95.1 ± 05.5	00.0 ± 00.0	04.1 ± 05.2	00.8 ± 03.2
WINE	95.7 ± 04.5	00.3 ± 01.0	03.5 ± 04.6	00.5 ± 01.2
BPMN	95.7 ± 05.6	01.0 ± 02.6	02.2 ± 04.6	01.0 ± 03.6
BIOPAX	97.4 ± 03.2	02.4 ± 03.1	00.0 ± 00.0	00.1 ± 00.5
NTN	93.0 ± 10.4	00.0 ± 00.0	03.6 ± 05.9	03.4 ± 09.9
SWSD	98.9 ± 02.0	00.0 ± 00.0	01.1 ± 02.0	00.0 ± 00.0
FINANCIAL	98.7 ± 02.6	00.6 ± 01.4	00.2 ± 01.0	00.5 ± 01.3
LUBM	99.0 ± 01.2	00.0 ± 00.0	01.0 ± 01.2	00.0 ± 00.0

Evaluation indices: The indices adopted have been used also in various papers on similar methods, (e.g. (d'Amato et al., 2008b)) compare the predictions made on test instances using the induced model to labels decided by a standard DL reasoner[14] for OWL used to provide a baseline.

For the evaluation, differently from other works that adopt the standard precision/recall measures which fail to take into account the missing knowledge cases due to the specific semantics of the data, other indices have been used (d'Amato et al., 2006b, 2008b) that, unlike the previous ones, do not have a direct mapping to the sets of true/false positives/negatives:

- *match*: case of an individual that got the same label by the reasoner and the inductive classifier;
- *omission error*: case of an individual for which the inductive method could not determine whether it was relevant to the query or not (response 0) while it was found relevant by the reasoner (response ± 1);
- *commission error*: case of an individual found to be relevant to the query concept (response -1 or $+1$) by the inductive classifier, while it logically belongs to its negation or vice-versa (response $+1$ or -1, respectively);
- *induction*: case of an individual found to be relevant to the query concept or to its negation (response ± 1), while either case is not logically derivable from the knowledge base (response 0).

A ten-fold cross validation design was adopted for the repetition of the experiments. The outcomes have been averaged over the target concepts.

Evaluation Results: The average index rates and standard deviations per each ontology are reported in Tab. 1. It is important to note that, for every ontology, the commission error was null. This means that the inductive model did not make critical mistakes, i.e. cases when an individual is deemed as an instance of a concept while it really is an instance of another disjoint concept. At the same time it is important to note that very high match rates were registered for each ontology. Particularly, it is interesting to observe that the match rate increases with the increase of the number of individuals in the considered ontology. This is because, with statistical methods, the performance is likely to improve with the availability of large numbers of training examples, that means that there is more information for better separating the example space. A conservative behavior has also been observed, indeed the omission error rate was not null (although it was very low). This was probably due to a high number of training examples with an unknown membership w.r.t. certain concepts.

[14] We used PELLET: `http://clarkparsia.com/pellet`

Table 2 Results with the k-NN procedure applied to instance-checking problems on the same datasets.

ontology	match	commission	omission	induction
SWM	67.4 ± 10.3	00.0 ± 00.0	18.4 ± 06.5	14.2 ± 08.7
WINE	74.1 ± 07.4	00.2 ± 00.4	16.7 ± 03.8	08.9 ± 05.6
BPMN	57.5 ± 03.2	00.0 ± 00.0	22.2 ± 23.1	00.3 ± 01.2
BIOPAX	73.7 ± 08.1	00.0 ± 00.0	09.6 ± 02.6	16.6 ± 08.3
NTN	63.8 ± 05.0	00.0 ± 00.0	20.6 ± 03.0	15.6 ± 03.9
SWSD	65.2 ± 14.2	00.0 ± 00.0	18.2 ± 07.3	16.5 ± 11.0
FINANCIAL	63.1 ± 02.5	01.5 ± 00.4	26.0 ± 01.9	09.4 ± 01.6
LUBM	63.5 ± 04.6	00.0 ± 00.0	16.7 ± 02.8	19.9 ± 03.5

To decrease the tendency to a conservative behavior of the method, a threshold could be introduced for the consideration of the "unknown" training examples. In almost any case the inductive model was able to induce some new knowledge. However, the assessment of the quality of the induced knowledge is not possible because the correct answer to the inferred membership assertions is known by the experts that built and populated the ontologies. Finally, one may also observe that the method had quite a stable behavior as testified by the limited deviations reported in the tables.

For a comparison with the k-Nearest Neighbor procedure, based on the proposed pseudo-metrics (see Sect. 3.1), the experiments were repeated with the same settings. The outcomes of these experiments – see Tab. 2 – clearly show the effectiveness of the models learned by the SVMs, as a mass of cases moves from the match case to omission error rates, which is only partially compensated by the number cases of induction. The k-Nearest Neighbor appears robust w.r.t. commission error cases. Even more so, a higher variance is also observed, denouncing an inferior stability of the algorithm.

From a qualitative viewpoint, using an alternative classification model emerging from *ABox* assertions one may be able to to detect constraints that have not been logically coded. E.g. considering the WINE ontology, a wine that is a `KathrynKennedyLateral` is also known to be a `Meritage` (super-class) yet not a `CaliforniaWine` (or an `AmericanWine`) as this is not logically derivable.

8.1.2 Property Value Prediction

We conducted experiments on a second classification task by predicting the value of specific properties for given entities. Consider for example a data set describing a set of people. One problem in this setting may be that some people have no age associated. The goal of the prediction task thus is to associate people with the class corresponding to their age.

In the evaluation, we compare the kernel functions based on intersection graphs and intersection trees as described in Sect. 4.2 to graph kernels in the application on an RDF dataset.

Data Set: Experiments are conducted on a Friend-of-a-Friend (FOAF) data set extracted from LiveJournal. The purpose of the FOAF project is to create a web of machine-readable pages describing people, their relationships, and people's activities and interests, using W3C's RDF technology. The FOAF ontology is based on RDFS/OWL and is formally specified in the FOAF Vocabulary Specification 0.91[15].

[15] `http://xmlns.com/foaf/spec/`

Table 3 Number of known instances of classes and number of known instances of properties in the FOAF dataset.

Class	#Inst.	Properties	#Inst.
$Location$	1344	$located$	3735
$School$	2794	$attend$	4118
$ChatAccount$	5	$holdsOnlineAccount$	3008
$Person$	22745	$knows$	116023
		$hasImage$	4554
$Date$	4	$dateOfBirth$	1567
$\#BlogPosts$	5	$posted$	4872

Table 4 Evaluation result for kernels based on intersection trees on the FOAF dataset - kernels are normalized

Kernel configuration				EvaluationResults			
Structure	inst. depth	maxSize	Discount	Error	Precision	Recall	F1 measure
Shervashidze and Borgwardt (2009)	2	2		0.2215	0.4174	0.3998	0.4084
Gärtner et al. (2003)	2		0.5	0.3824	**0.4841**	**0.5964**	**0.5344**
Paths	2	2	1	**0.1946**	0.4785	0.3729	0.4192
Full Subtree	2		1	0.2312	0.3986	0.3875	0.3930

Descriptions of 22745 people and their properties are extracted from the LiveJournal website. The total number of instances of the describing entity classes and properties are shown in Table 3. Note that the instances of classes $OnlineChatAccount$, $Date$ and $\#BlogPosts$ are summarized to a small number of discrete states. Of the 22745 described people, 1567 are associated with a date of birth. These persons are used as instances in the classification task. In the evaluation, the goal consisted in predicting for each of the 1567 people with an age information to which of the four age classes they belonged.

Evaluation Setting: Evaluation results are obtained using leave-one-out Cross Validation on the whole data set. Accuracy, precision, recall and F-measure are used as evaluation measures. Support Vector Machines were learnt using the kernel functions presented in Sect. 4.2.

Evaluation Results: We report results of the learned classifiers in Table 4. The results show that in general graph kernels do not perform well on RDF data. The path kernel which is obtained by counting paths up to length 2 in the intersection tree can outperform classical graph kernels and also the Full Subtree kernel in terms of classification errors. The Full Subtree kernel is obtained by assessing the size of the intersection tree in terms of number of nodes. However, the intersection tree can be calculated faster than the intersection graph which is the basis for the Paths kernel. The Gaertner kernel outperforms the other kernel functions in terms of precision, recall and F-measure. However, while the results for the other kernels were obtained within hours, it took more than 6 weeks to obtain the classification results of the Gaertner kernel. The calculation of the Gaertner kernel is that expensive as each kernel calculation requires the inversion of the adjacency matrix of the intersection graph.

8.2 Relation Prediction

In the next section we report results on basic link prediction before presenting an extension that can incorporate more expressive ontological constructs (see Sec. 8.2.2).

8.2.1 Link Prediction

This section summarizes an application of multivariate structured prediction algorithms like the ones discussed in Sec. 5 to predict the existence of links between to instances in a social network. This could be e.g. the *knows*-relation between to concrete persons. In the context of the SW this could be used to assess the certainty that two persons might know each other. Huang et al. (2010) and Kiefer et al. (2008) proposed extension to SW query languages like SPARQL to query such learned probabilistic statements from a SW knowledge base. Different crawling an sampling strategies are investigate to assess the performance under various real world setups. For details on the algorithm and experiments see (Huang et al., 2010).

Data Set: As before, the experiments are based on friend-of-a-friend (FOAF) data. All extracted entities and relations are shown in Figure 1. In total 32,062 persons and all related attributes are collected. From this triple set, which is called "full triple set", 14,425 persons with a 'dense' friendship information are extracted. On average, a given person has 27 friends. Then, rare attributes which are associated with less than 10 persons are pruned. Table 5 lists the number of different individuals (top rows) and their known instantiated relations (bottom rows) in the full triple set, in the pruned triple set and in triples sets in different experiment settings (explained below). Note that $OnlineChatAccount$, $Date$ and $\#BlogPosts$ are reduced to a small number of discrete states. The resulting data matrix, after pruning, has 14,425 rows (persons) and 15,206 columns. Among those columns 14,425 ones (friendship attributes) refer to the property *knows* (see Figure 1). The remaining 781 columns (general attributes) refer to general information about age, location, number of blog posts, attended school, online chat account and interest.

Data Retrieval and Sampling Strategies: In the experiments the generalization capabilities of the learning algorithms given 4 different situations is evaluated.

Setting 1 describes the situation where the depicted part of the SW is randomly accessible, meaning that all instances can be queried directly from the triple stores. Statistical units in the sample for training are randomly sampled and statements for other randomly selected statistical units are predicted for testing (inductive setting). This way, on average persons are barely connected by the $knows$ relation. The $knows$ relation in the training and test set are very sparse (0.18%).

Setting 2 also shows the situation where statistical units in the sample are randomly selected, but this time the truth values of statements concerning the statistical units in the training sample are predicted (transductive setting). Instances of the $knows$ relation are withheld from training and used for prediction. Prediction should be easier here since the statistics for training and prediction match perfectly.

Setting 3 assumes that the Web address of one user (i.e., statistical unit) is known and only a subpart of the data can be collected by crawling, which is a typical situation on the Web. Starting from this random user profile, the profiles of users connected by the $knows$ relation are gathered by crawling breadth-first and are then added to the training set. The test set is gathered by continued crawling symbolized by the outer circle (inductive setting). This way all profiles are (not necessarily directly) connected and training profiles show a higher connectivity (1.02%) compared to test profiles (0.44%). In this situation generalization can be expected to be easier because local properties are more consistent than global ones.

Table 5 Number of individuals and number of instantiated relations in the full triple set, in the pruned triple set (see text) and statistics for the different experimental settings

		full	pruned	setting 1	setting 2	setting 3	setting 4
Concept	$Person$	32,062	14,425	4,000	2,000	4,000	2,000
#Indivi.	$Location$	5,673	320	320	320	320	320
	$School$	15,744	329	329	329	329	329
	$Interest$	4,695	118	118	118	118	118
	$On.ChatAcc.$	5	5	5	5	5	5
	$Date$	4	4	4	4	4	4
	$\#BlogPosts$	5	5	5	5	5	5
Role	$knows$	530,831	386,327	14,650	7,339	58,399	40,786
#Inst.	(sparsity)	(0.05%)	(0.19%)	train(0.18%)	(0.18%)	train (1.02%)	(1.02%)
				test (0.18%)		test (0.44%)	
	$residence$	24,368	7,964	2,228	1,106	2,389	1,172
	$attends$	31,507	5,088	1,423	747	1,467	718
	has	9,607	1,645	449	245	420	214
	$holds$	19,021	8,319	2,221	1,087	2,243	1,168
	$dateOfBirth$	10,040	5,287	1,492	715	1,563	779
	$posted$	31,959	14,369	3,985	1,992	3,985	1,994

Setting 4 is the combination of setting 2 and 3. The truth values of statements concerning the statistical units in the training sample are predicted (transductive setting). Instances of the $knows$ relation are withheld from training and used for prediction.

Evaluation Procedure and Evaluation Measure: The task is to predict potential friends of a person, i.e., $knows$ statements. For each person in the data set, one $knows$ friendship statement is randomly selected and the corresponding matrix entry is set to $zero$, to be treated as unknown (test statement). In the test phase then all unknown friendship entries are predicted, including the entry for the test statement. The test statement should obtain a high likelihood value, if compared to the other unknown friendship entries. Here, the normalized discounted cumulative gain (NDCG) (Jarvelin and Kekalainen, 2000) is used to evaluate a predicted ranking. The better the algorithm, the higher the friendship test statement would be ranked.

Benchmarked methods: Besides reduced rank penalized regression (RRPP) we investigate matrix completion based on a singular value decomposition (SVD), matrix completion based on non-negative matrix factorization (NNMF) (Lee and Seung, 1999) and matrix completion using latent Dirichlet allocation (LDA) (Blei et al., 2003). All approaches estimate unknown matrix entries via a low-rank matrix approximation. As simple baseline methods we report: *Baseline:* Here, a random ranking for all unknown triples is assumed, i.e., every unknown triple gets a random probability assigned. *Friends of friends in second depth (FOF, d=2):* It is assumed that friends of friends of a particular person might be friends of that person too, which means that the next indirect friendship relations is taken as direct friendships and all other possible friendships are ignored. From the RDF graph point of view the $knows$ relation propagates one step further alongside the existing $knows$ linkages.

Results: In settings 1 and 2, 2,000 persons are randomly sampled for the training set. In addition, in setting 1, 2,000 further persons are randomly sampled for the test set. In setting 3, 4,000 persons were sampled, where the first half were used for training and the second half for testing. Setting 4 only required the 2,000 persons in the training set. In each case,

Table 6 Best *NDCG all* and standard error where z stands for the number of latent variables

Method	setting 1	setting 2	setting 3	setting 4
$Baseline$	0.1092 ± 0.0003	0.1092 ± 0.0003	0.1094 ± 0.0001	0.1094 ± 0.0001
$FOF, d = 2$	0.2146 ± 0.0095	0.2146 ± 0.0095	0.1495 ± 0.0077	0.1495 ± 0.0077
$NNMF$	NaN	0.2021 ± 0.0058 z=100	NaN	0.2983 ± 0.0197 z=150
SVD	0.2174 ± 0.0061 z=150	0.2325 ± 0.0074 z=100	0.2085 ± 0.0147 z=200	0.3027 ± 0.0179 z=100
LDA	0.2512 ± 0.0049 z=200	0.2988 ± 0.0057 z=200	0.2375 ± 0.0123 z=200	0.3374 ± 0.0117 z=200
$RRPP$	0.2483 ± 0.0018 z=400	0.2749 ± 0.0037 z=400	0.2252 ± 0.0049 z=400	0.3315 ± 0.0109 z=400

sampling was repeated 5 times such that error bars could be derived. Table 5 reports details of the samples (training set and, if applicable, test set). The matrix completion methods introduced in Section 5 were then applied to the training set.

For each sample the evaluation procedure described above was repeated 10 times, i.e., random selection of one $knows$ relation per person was treated as unknown. Since NNMF is only applicable in a transductive setting, it was only applied in setting 1 and 3.

The best *NDCG all* scores of all algorithms in different settings are shown in Table 6, where z indicates the number of latent variables when the best scores are achieved. Comparing the results over different settings it can be found that for three matrix completion methods one obtains best performance in setting 4, next best performance in setting 2, then follows setting 1 and setting 3 is the most difficult. The baseline method, random guess, is independent to the settings and achieves almost the same score. A single irregularity is that FOF, d=2 in setting 2 performs better than in setting 4.

The fact that the scores in setting 4 are the best indicates that a link-following sampling strategy increases indeed the performance of learning methods. On one side, the sampled persons are more likely to come from same communities and have similar profiles so that they are more likely to know each other. On the other side the $knows$ relation is more dense than the case of random sampling (see Table 5). In the latter case persons more rarely have common friends. The experimental results confirm the assumption that the more sparse the matrix is, the more difficult the problem becomes since friendship patterns are more rare. In addition, we observe that the prediction performance in setting 1 is not much worse than the prediction performance in setting 2. Although from disjoint sets the statistics in training and testing is similar, leading to comparable results. Interestingly, it can be seen that the performance of setting 3 is much worse that the prediction in setting 4. We attribute this to the general statistics in the training and the test set because they are very different in setting 3. In Table 5 it is apparent that in setting 3 the $knows$ relation in the training data set (1.02%) is significantly more dense than in the test data set (0.44%). Intuitively speaking, the people in the training know each other quite well, but the people in the test do not know the people in the training as much.

In general, we observe that LDA and RRPP outperform NNMF and SVD in each setting. In addition, these two methods are not sensitive to the predefined number of latent variables as long as the chosen number is reasonably high. LDA reaches its maximum NDCG score, for instance, with $z = 150$ latent variables in settings 4 and the performance does not deteriorate when the number of latent factors is increased. The score of RRPP keeps increasing and does not drop down either. In contrast, NNMF and SVD are sensitive with respect to the predefined number of latent variables. NNMF reaches the maximum with $z = 150$ and

Table 7 Number of individuals, no. of instantiated roles in the reduced LJ-FOAF data set

Concept	#Individuals	Role	#Instances
$Location$	200	$residence$	514
$School$	747	$attends$	963
$OnlineChatAccount$	5	$holdsAccount$	427
$Person$	638	$knows$	8069
		$hasImage$	574
$Date$	4	$dateOfBirth$	194
$\#BlogPosts$	5	$posted$	629

Table 8 Examples of constraints used in the FOAF-ontology represented in OWL DL ($\mathcal{SHOIN}(D)$).

$$\begin{aligned} Pupil &\sqsubseteq Person \\ Pupil &\sqsubseteq \neg UnderSixOld \\ Pupil &\sqsubseteq \exists\, attendsSchool \end{aligned}$$

$z = 100$ in setting 4 , while the highest score of SVD occurs by $z = 100$ and $z = 50$ in the same setting.

This approach can be extended in many ways. One extension concerns ontological background knowledge. So far, ontological background knowledge was considered by including logically inferred statements into learning. In the next section experiments are shown using an approach that can incorporate expressive DL constructs.

8.2.2 Constrained Relation Prediction

In this section we summarize an application of the latent-class relational graphical models IHRM and IHSM as introduced in Sec. 6.3 and Sec. 7.1.2 to relation prediction in expressive DL ontologies. To incorporate DL axioms specified in the ontology, e.g., in the form of hard constraints in the learning process, DL consistency checks are constantly performed during Gibbs sampling iterations using a tableau-based algorithm. For details on the algorithm and experiments see also (Rettinger et al., 2009).

Implementation and Data: While data mining techniques can handle large amounts of simple facts, little effort has been made to exploit the semantic information inherent in social networks while incorporating constraints specified by system and user profile designers. This section presents both, a large and complex SW dataset containing DL axioms and the methodology of how to apply ML in the form of the IHSM in practice.

The data set used is a subset of the data set shown in Tab. 3. Tab. 7 lists the number of different individuals (left column) and their known instantiated roles (middle column) used for experiments. This real world data set offers both, a sufficiently large set of individuals for inductive learning and a formal ontology, LJ-FOAF, specified in RDFS and OWL. However, while LJ-FOAF offers a taxonomy, no complex constraints are given. Thus, to demonstrate the full potential of IHSM, the constraints in Tab. 8 were added: no one who is younger than six years goes to school (see Tab. 8). $UnderSixYearsOld$ is the class which contains persons with an age less than six years (calculated from the given dates of birth).

To implement all features of IHSM the following open source software packages are used in the presented application: Protege[16] is used to adjust the FOAF ontology to OWL DL and to implement additional axioms. The SW framework Jena[17] is used to load, store

[16] `http://protege.stanford.edu/`
[17] `http://jena.sourceforge.net/`

Table 9 Predictive performance for different LJ-FOAF roles: AUC and 95% confidence intervals

Relation Class	attends	dateOfBirth	knows
IHRM	0.577 ($\pm$0.013)	0.548 ($\pm$0.018)	0.813 ($\pm$0.005)
IHRM+C	0.581 ($\pm$0.012)	0.549 ($\pm$0.016)	0.814 ($\pm$0.006)
IHSM	**0.608** ($\pm$0.017)	**0.561** ($\pm$0.011)	**0.824** ($\pm$0.002)

and query the ontology and Pellet[18] provides the OWL DL reasoning capabilities. The Gibbs sampling procedure is implemented in Java with the help of Colt[19], an open source library for high performance scientific and technical computing.

Here is an outline of the workflow: First, the *TBox* axioms are designed and loaded into Jena. Next, all *ABox* assertions are added and loaded into Jena. Then, by using the taxonomy information from the ontology and the *ABox* assertions, a Relational Model (RM) is extracted. This RM is transferred into an IHSM by adding hidden variables and parameters, accordingly. Finally, the parameters are learned from the data, while constraints are constantly checked by the DL reasoner. The trained model can now be used for statistical analysis, like prediction of unknown relation instances.

In the experiments the standard reported setting for the truncation parameter is $\#Individuals/10$ for entity classes with over 100 instances and $\#Individuals$ for entity classes with less individuals. The standard iterations of the Gibbs sampler are 2000. $\alpha_0 = 5$ is fixed for every entity class and $\beta_0 = 20$ for every relation class.

Evaluation Procedure and Evaluation Measure: In social network analysis one could for instance want to predict 'who knows who' in case either this information is unknown or the systems wants to recommend new friendships. Other relations that could be interesting to predict in case they are unknown are the school someone attends/attended or the place he lives/lived. Furthermore, one could want to predict unspecified attributes of certain persons, like their age. For a broader coverage of social network analysis tasks see e.g. (Mika, 2004).

The comparison of IHSM to other first-order probabilistic learning algorithms (see Sec. 7) is difficult due to the fact that only IHSM uses hard constraints. Thus, the influence of the constraining on the results of the learning process is shown, as it was examined by Punyakanok et al. (2005). In particular, the influence of constraining on the predictive performance for IHSM compared to IHRM is shown. In addition, the performance of IHRM with a subsequent constraining of the predictions is given. In this case, inconsistent predictions are also avoided, however the learned model remains the same. This setup is denoted IHRM+C.

A 5-fold cross validation is performed to evaluate the predictions of different relation classes. In specific, the non-zero entries of the relation matrix to be predicted were randomly split in 5 parts. Each part was once used for testing while the remaining parts were used for training. The entries of each testing part were set to zero (unknown) for training and to their actual value of 1 for testing. Each fold was trained with 1000 iterations of the Gibbs sampler, where 500 iterations are discarded as the burn-in period. After this, the learned parameters are recorded every 50th iteration. In the end we use the 10 recorded parameter sets to predict the unknown relation values, average over them and calculate the area under the ROC curve (AUC) as our evaluation measure. Finally we average over the 5 folds and calculate the 95% confidence interval.

[18] `http://pellet.owldl.com/`

[19] `http://acs.lbl.gov/~hoschek/colt/`

Results: The obvious roles to evaluate are $attends$ and $dateOfBirth$. Both are constrained by the ontology, so IHSM should have an advantage over IHRM because it cannot predict any false positives. The results in Table 9 confirm this observation. In both cases IHSM did outperform IHRM. Interestingly, IHRM+C only gives a slightly improved performance. This is due to the fact, that still an unconstrained model is learned. Thus, only a few inconsistent predictions are avoided, but no global model consistent with the constraints is learned. This confirms the results of Punyakanok et al. (2005).

A less obvious outcome can be examined from the influence of the constraining on a relation that is not directly constrained by the ontology like $knows$. Still, in those experiments IHSM shows a slight advantage over IHRM. Thus, there seems to be a positive influence of the background knowledge, although a lot of users specify an incorrect age. However, there is the potential that the opposite may occur likewise. If the given constraints are conflicting with the empirical evidence there could even be a decrease in predictive performance. Ultimately, it is the ontology designers choice to decide whether to enforce a constraint that conflicts with the observed evidence.

8.3 Clustering

Clustering of instances of each entity class can provide valuable insights of the domain under investigation. An example is clustering of RDF-instances of gene annotations for summarization of the RDF-graph (Thor et al., 2011). For the area of social network analysis this could mean finding similar persons to identify sub-groups of people. Those clusters can then e.g., be used for link prediction or adding more specific entity classes to the taxonomy.

Implementation, Data and Experiments: In this section we will report experimental results concerning the task of clustering users in a social network. The approach applied is IHRM as introduced in Sec. 6.3 which estimates unknown relations by clustering individuals in each entity class in a global unsupervised optimization procedure. Note that IHRM determines the relevance of each relation type and the number of components for each entity class in an unsupervised fashion and estimates unknown values at the same time.

Thus, for the clustering task the same implementation, data set and experimental runs reported for the link prediction task in Sec. 8.2.2 apply to clustering. By investigating the result of the latent classes learned by the algorithm we can find interesting information about the domain. Recall that we tested an extension to IHRM, called IHSM, which incorporates DL axioms specified in the ontology, e.g., in the form of hard constraints into the learning process.

Results: Table 10 shows the number of components found for each entity class. In this setup of experiments the main factor which influenced the clustering of persons turned out to be the age of each user. One interesting outcome of the comparison of IHRM and IHSM is the number of components per hidden variable after convergence (see Table 10 right column). In both cases, if compared to the initialization, the algorithms converged to a much smaller number of components. Most of the individuals were assigned to a few distinct components leaving most of the remaining components almost empty.

However, there is a noticeable difference between IHRM and IHSM concerning the concepts $School$ and $Person$ which needed more components after training with IHSM (see bold numbers in Table 10). A closer analysis of the components revealed that IHSM generated additional components for inconsistent individuals, because both concepts are

Table 10 Number of components found for each entity class in the social network.

Entity Class	Relation Class	#Compo. IHRM	#Compo. IHSM
$Location$	$residence$	18	17
$School$	$attends$	**36**	**48**
$OnlineChatAccount$	$holdsAccount$	4	4
$Person$	$knows$	**38**	**45**
$Date$	$dateOfBirth$	**4**	**2**
$\#BlogPosts$	$posted$ 4	4	

affected by constraints. Inconsistent individuals are in this example persons that specified both, being under the age of 6 and attending a school.

In contrast, the last concept affected by the constraints ($Date$) shows an opposite effect if constraining is used: the results show fewer components. Here, IHSM divided more generally into age groups 'too young' and 'old enough' which also reflects the constraints.

All things considered, this demonstrates that the restriction of roles in form of consistency checks does influence the states of the latent variables. In this example the changes to the states appear to be intuitive.

Note, that the learned clusters can also be used to extract meta-knowledge from the data, similar to latent topics that are extracted from documents in LDA (Bundschus et al., 2009). The lower dimensional latent space can be used to define more general concepts. This abstraction can be used to simplify reasoning and inference. How this latent structure could be transformed into symbolic knowledge and fed back to the ontology is a promising direction of future research.

9 Concluding Remarks

Advanced Machine Learning (ML) techniques show the potential to automate a number of relevant tasks for the Semantic Web (SW) by complementing and integrating logical inference with inductive procedures that exploit regularities in the data. An obvious benefit of inductive methods is that they are more robust against some of the inherent problems of the SW such as contradicting information, incomplete information and non-stationarity. In Sec. 2, we introduced the most common SW knowledge representation and related ML tasks.

In Sec. 3-7, this paper surveyed statistical learning solutions to cope with problems in the context of typical representations for the Semantic Web. We presented similarity-based methods, such as nearest-neighbor approaches, multi-layer networks and kernel machines, which may readily be applied to SW data. Then, we moved to a discussion of ML algorithms specialized on representations underlying the Internet and specifically the SW. Multivariate prediction models and relational graphical models allow for the application of statistical inference like predictions or clustering to such knowledge bases. We conclude the algorithmic sections with an introduction to the most expressive ML models which combine logical and probabilistic REPRESENTATIONS and enable deductive and inductive reasoning.

In Sec. 8, we presented the outcome of our empirical evaluations of some of the presented techniques on real-world SW-mining tasks like the analysis of social networks represented in SW formats. Here we focused on classification and prediction of (unknown) instances and relations in the ontology, while taking into account specifics of the crawling strategy to gather the data from the SW. We also considered data sets of varying expressivity. Additional tasks like clustering and query answering are briefly discussed.

One lesson learned from our experiments with this wide spectrum of SW-related algorithms and tasks is that it is not feasible to find the one single best algorithm for SW mining tasks in general. First, SW data sets have strongly varying characteristics depending on the domain they are describing. Some algorithms perform well on social network data, others are stronger on scientific data, like biomedical gene-disease networks. Second, the same differences were observed for varying crawling strategies. Depending on the sampling of the data points from the SW the algorithms performed differently. Finally, there are many algorithms that are limited by the expressivity of the knowledge representation and the size of the data set that they can handle. Some are not suited for highly expressive logical representation, while others struggle with large data sets containing millions of facts.

With the growing amounts of data becoming available in semantic representations, the investigation of mining techniques for the Semantic Web constitutes an emerging branch of Web mining with specific problems and solutions, a field that promises to attract more interest in the near future.

References

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P., editors (2003). *The Description Logic Handbook*. Cambridge University Press.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*.

Bicer, V., Tran, T., and Gossen, A. (2011). Relational kernel machines for learning from graph-structured rdf data. In Antoniou, G. et al., editors, *Proceedings of the 8th Extended Semantic Web Conference, ESWC 2011*, volume 6643 of *LNCS*, pages 47–62. Springer.

Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data–the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Bloehdorn, S., Haase, P., Sure, Y., and Voelker, J. (2006). Ontology evolution. In Davies, J., Studer, R., and Warren, P., editors, *Semantic Web Technologies*. Wiley.

Bloehdorn, S. and Sure, Y. (2007). Kernel methods for mining instance data in ontologies. In Aberer, K. and et al., editors, *Proceedings of the 6th International Semantic Web Conference, ISWC2007*, volume 4825 of *LNCS*, pages 58–71. Springer.

Bock, H. (1999). *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*. Springer-Verlag.

Borgida, A., Walsh, T., and Hirsh, H. (2005). Towards measuring similarity in description logics. In Horrocks, I., Sattler, U., and Wolter, F., editors, *Working Notes of the International Description Logics Workshop*, volume 147 of *CEUR Workshop Proceedings*, Edinburgh, UK.

Brickley, D. and Miller, L. (2007). FOAF vocabulary specification. Technical report, FOAF project. Published online on May 24th, 2007 at `http://xmlns.com/foaf/spec/20070524.html`.

Bright, M. W., Hurson, A. R., and Pakzad, S. H. (1994). Automated resolution of semantic heterogeneity in multidatabases. *ACM Transaction on Database Systems*, 19(2):212–253.

Buitelaar, P., Olejnik, D., and Sintek, M. (2004). A protege plug-in for ontology extraction from text based on linguistic analysis. In *Proceedings of the 1st European Semantic Web Symposium(ESWS)*.

Bundschus, M., Yu, S., Tresp, V., Rettinger, A., Dejori, M., and Kriegel, H.-P. (2009). Hierarchical bayesian models for collaborative tagging systems. In *IEEE International Conference on Data Mining series (ICDM 2009)*.

Carbonetto, P., Kisynski, J., de Freitas, N., and Poole, D. (2005). Nonparametric bayesian logic. In *Proc. 21st UAI*.

Cimiano, P., Hotho, A., and Staab, S. (2005). Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research (JAIR)*, 24:305–339.

Cimiano, P. and Völker, J. (2005). Text 2 onto-a framework for ontology learning and data-driven change discovery.

Cohen, W. and Hirsh, H. (1994). Learning the CLASSIC description logic. In Torasso, P., Doyle, J., and Sandewall, E., editors, *Proceedings of the 4th International Conference on the Principles of Knowledge Representation and Reasoning*, pages 121–133. Morgan Kaufmann.

Cumby, C. and Roth, D. (2003). On kernel methods for relational learning. In Fawcett, T. and N.Mishra, editors, *Proceedings of the 20th International Conference on Machine Learning, ICML2003*, pages 107–114. AAAI Press.

da Costa, P., d'Amato, C., Fanizzi, N., Laskey, K., Laskey, K., Lukasiewicz, T., Nickles, M., and Pool, M., editors (2008). *Uncertainty Reasoning for the Semantic Web I*, volume 5327 of *LNCS*. Springer.

d'Amato, C., Fanizzi, N., and Esposito, F. (2005). A semantic similarity measure for expressive description logics. In Pettorossi, A., editor, *Proceedings of Convegno Italiano di Logica Computazionale (CILC05)*, Rome, Italy. `http://www.disp.uniroma2.it/CILC2005/downloads/papers/15.dAmato_CILC05.pdf`.

d'Amato, C., Fanizzi, N., and Esposito, F. (2006a). A dissimilarity measure for $\mathcal{ALC}$ concept descriptions. In *Proceedings of the 21st Annual ACM Symposium of Applied Computing, SAC2006*, volume 2, pages 1695–1699, Dijon, France. ACM.

d'Amato, C., Fanizzi, N., and Esposito, F. (2006b). Reasoning by analogy in description logics through instance-based learning. In Tummarello, G., Bouquet, P., and Signore, O., editors, *Proceedings of Semantic Web Applications and Perspectives, 3rd Italian Semantic Web Workshop, SWAP2006*, volume 201 of *CEUR Workshop Proceedings*, Pisa, Italy.

d'Amato, C., Fanizzi, N., and Esposito, F. (2008a). Analogical reasoning in description logics. In da Costa, P. et al., editors, *Uncertainty Reasoning for the Semantic Web I*, volume 5327 of *LNAI*, pages 336–354. Springer.

d'Amato, C., Fanizzi, N., and Esposito., F. (2008b). Query answering and ontology population: An inductive approach. In Bechhofer, S. and et al., editors, *Proceedings of the 5th European Semantic Web Conference, ESWC2008*, volume 5021 of *LNCS*, pages 288–302. Springer.

d'Amato, C., Staab, S., and Fanizzi, N. (2008c). On the influence of description logics ontologies on conceptual similarity. In Gangemi, A. and Euzenat, J., editors, *Proceedings of the 16th EKAW Conference, EKAW2008*, volume 5268 of *LNAI*, pages 48–63. Springer.

De Raedt, L. (2008). *Logical and Relational Learning: From ILP to MRDM (Cognitive Technologies)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

De Raedt, L., Frasconi, P., Kersting, K., and Muggleton, S., editors (2008). *Probabilistic Inductive Logic Programming - Theory and Applications*, volume 4911 of *Lecture Notes in Computer Science*. Springer.

De Salvo Braz, R., Amir, E., and Roth, D. (2005). Lifted first-order probabilistic inference. In *IJCAI'05: Proceedings of the 19th international joint conference on Artificial intelligence*, pages 1319–1325, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ding, L., Kolari, P., Ding, Z., and Avancha, S. (2007). Using Ontologies in the Semantic Web: A Survey. 14:79–113.

Ding, Z. (2005). *BayesOWL: A Probabilistic Framework for Semantic Web*. PhD thesis, University of Maryland, Baltimore County.

Domingos, P. and Richardson, M. (2007). Markov logic: A unifying framework for statistical relational learning. In Getoor, L. and Taskar, B., editors, *Introduction to Statistical Relational Learning*. MIT Press.

Duda, R., Hart, P., and Stork, D. (2001). *Pattern Classification*. Wiley, 2nd edition.

Euzenat, J. and Shvaiko, P. (2007). *Ontology matching*. Springer.

Fanizzi, N. and d'Amato, C. (2006). A declarative kernel for $\mathcal{ALC}$ concept descriptions. In Esposito, F. and et al., editors, *Proceedings of the 16th International Symposium on Methodologies for Intelligent Systems, ISMIS2006*, volume 4203 of *LNAI*, pages 322–331. Springer.

Fanizzi, N. and d'Amato, C. (2007). Inductive concept retrieval and query answering with semantic knowledge bases through kernel methods. In Apolloni, B., Howlett, R., and Jain, L., editors, *Proceedings of the 11th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, KES2007*, volume 4692 of *LNAI*, pages 148–155. Springer.

Fanizzi, N., d'Amato, C., and Esposito, F. (2007). Induction of optimal semi-distances for individuals based on feature sets. In Calvanese, D. and et al., editors, *Working Notes of the 20th International Description Logics Workshop, DL2007*, volume 250 of *CEUR Workshop Proceedings*, Bressanone, Italy.

Fanizzi, N., d'Amato, C., and Esposito, F. (2008a). DL-FOIL: Concept learning in Description Logics. In Zelezný, F. and Lavrač, N., editors, *Proceedings of the 18th International Conference on Inductive Logic Programming, ILP2008*, volume 5194 of *LNAI*, pages 107–121, Prague, Czech Rep. Springer.

Fanizzi, N., d'Amato, C., and Esposito, F. (2008b). Evolutionary conceptual clustering based on induced pseudo-metrics. *Semantic Web Information Systems*, 4(3):44–67.

Fanizzi, N., d'Amato, C., and Esposito, F. (2008c). Learning with kernels in description logics. In Zelezný, F. and Lavrač, N., editors, *Proceedings of the 18th International Conference on Inductive Logic Programming, ILP2008*, volume 5194 of *LNAI*, pages 210–225. Springer.

Fanizzi, N., d'Amato, C., and Esposito, F. (2008d). Statistical learning for inductive query answering on OWL ontologies. In Sheth, A. and et al., editors, *Proceedings of the 7th International Semantic Web*

Conference, ISWC2008, volume 5318 of *LNCS*, pages 195–212. Springer.

Fanizzi, N., d'Amato, C., and Esposito, F. (2009). ReduCE: A reduced coulomb energy network method for approximate classification. In Aroyo, L. and et al., editors, *Proceedings of the 6th European Semantic Web Conference, ESWC2009*, volume 5554 of *LNCS*, pages 323–337. Springer.

Gärtner, T., Flach, P., and Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. In Schölkopf, B. and Warmuth, M. K., editors, *Computational Learning Theory and Kernel Machines (COLT/Kernel 2003)*, volume 2777 of *Lecture Notes in Computer Science*, pages 129–143. Springer, Berlin–Heidelberg, Germany.

Gärtner, T., Lloyd, J., and Flach, P. (2004). Kernels and distances for structured data. *Machine Learning*, 57(3):205–232.

Getoor, L., Friedman, N., Koller, D., Pferrer, A., and Taskar, B. (2007). Probabilistic relational models. In Getoor, L. and Taskar, B., editors, *Introduction to Statistical Relational Learning*. MIT Press.

Getoor, L. and Taskar, B., editors (2007). *Introduction to Statistical Relational Learning*. MIT Press.

Giugno, R. and Lukasiewicz, T. (2002). P-$\mathcal{SHOQ}$(D): A probabilistic extension of $\mathcal{SHOQ}$(D) for probabilistic ontologies in the semantic web. In *JELIA '02: Proceedings of the European Conference on Logics in Artificial Intelligence*, pages 86–97, London, UK. Springer-Verlag.

Grobelnik, M. and Mladenic, D. (2006). Knowledge discovery for ontology construction. In Davies, J., Studer, R., and Warren, P., editors, *Semantic Web Technologies*. Wiley.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning – Data Mining, Inference, and Prediction*. Springer.

Heckerman, D., Geiger, D., and Chickering, D. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243.

Hitzler, P. and Vrandečić, D. (2005). Resolution-based approximate reasoning for OWL DL. In Gil, Y. and et al., editors, *Proceedings of the 4th International Semantic Web Conference, ISWC2005*, number 3279 in LNCS, pages 383–397, Galway, Ireland. Springer.

Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B., and Dean, M. (2004). Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21:79.

Horvth, T., Grtner, T., and Wrobel, S. (2004). Cyclic pattern kernels for predictive graph mining. In Kim, W., Kohavi, R., Gehrke, J., and DuMouchel, W., editors, *Proceedings of the 10t ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004), August 22-25, 2004, Seattle, WA, USA*, pages 158–167. ACM Press, New York, NY, USA.

Huang, Y., Tresp, V., Bundschus, M., and Rettinger, A. (2009). Scalable relational learning for sparse and incomplete domains. In *Proceedings of the International Workshop on Statistical Relational Learning (SRL-2009)*.

Huang, Y., Tresp, V., Bundschus, M., and Rettinger, A. (2010). Multivariate structured prediction for learning on semantic web. *Proc. of the 20th International Conference on Inductive Logic Programming (ILP 2010)*.

Huynh, T. N. and Mooney, R. J. (2011). Online structure learning for markov logic networks. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2011)*, volume 2, pages 81–96.

Iannone, L., Palmisano, I., and Fanizzi, N. (2007). An algorithm based on counterfactuals for concept learning in the semantic web. *Applied Intelligence*, 26(2):139–159.

Jaeger, M. (1997). Relational bayesian networks. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI)*.

Janowicz, K. (2006). Sim-DL: Towards a semantic similarity measurement theory for the Description Logic $\mathcal{ALCNR}$ in geographic information retrieval. In Meersman, R. and et al. et al., editors, *Proceedings of SeBGIS 2006, OTM Workshops*, volume 4278 of *LNCS*, pages 1681–1692. Springer.

Janowicz, K., Keßler, C., Schwarz, M., Wilkes, M., Panov, I., Espeter, M., and Bäumer, B. (2007). Algorithm, implementation and application of the sim-dl similarity server. In *Proceedings of GeoS 2007, 2nd International Conference on GeoSpatial Semantics.*, LNCS, pages 128–145. Springer.

Janowicz, K. and Wilkes, M. (2009). Sim-dl_a: A novel semantic similarity measure for description logics reducing inter-concept to inter-instance similarity. In *Proceedings of the 6th Annual European Semantic Web Conference (ESWC2009).*, volume 5554 of *LNCS*, pages 353–367. Springer.

Jarvelin, K. and Kekalainen, J. (2000). IR evaluation methods for retrieving highly relevant documents. In *SIGIR'00*.

Karatzoglou, A., Amatriain, X., Baltrunas, L., and Oliver, N. (2010). Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 79–86, New York, NY, USA. ACM.

Kersting, K. and De Raedt, L. (2001). Bayesian logic programs. Technical report, Albert-Ludwigs University at Freiburg.

Kiefer, C., Bernstein, A., and Locher, A. (2008). Adding data mining support to sparql via statistical relational learning methods. In *ESWC 2008*. Springer-Verlag.

Kifer, M. (2008). Rule interchange format: The framework. *Web Reasoning and Rule Systems*, pages 1–11.

Koller, D., Levy, A. Y., and Pfeffer, A. (1997). P-CLASSIC: A tractable probablistic description logic. In *AAAI/IAAI*, pages 390–397.

Koller, D. and Pfeffer, A. (1998). Probabilistic frame-based systems. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting andlabeling sequence data. In *International Conference on Machine Learning*.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*.

Lee, J., Kim, M., and Lee, Y. (1993). Information retrieval based on conceptual distance in is-a hierarchies. *Journal of Documentation*, 2(49):188–207.

Lehmann, J. (2009). DL-Learner: learning concepts in description logics. *Journal of Machine Learning Research (JMLR)*. To appear.

Lehmann, J. and Hitzler, P. (2008). A refinement operator based learning algorithm for the $\mathcal{ALC}$ description logic. In Blockeel, H., Ramon, J., Shavlik, J., and Tadepalli, P., editors, *Proceedings of the 17th International Conference on Inductive Logic Programming, ILP2007*, volume 4894 of *LNCS*. Springer.

Lippert, C., Huang, Y., Weber, S. H., Tresp, V., Schubert, M., and Kriegel, H.-P. (2008). Relation prediction in multi-relational domains using matrix factorization. Technical report, Siemens.

Lisi, F. A. and Esposito, F. (2005). An ilp perspective on the semantic web. In *Proceedings of SWAP 2005, the 2nd Italian Semantic Web Workshop, Trento, Italy, December 14-16, 2005, CEUR Workshop Proceedings*.

Lukasiewicz, T. (2007). Probabilistic description logic programs. *Int. J. Approx. Reasoning*, 45(2):288–307.

Maedche, A. and Staab, S. (2004). Ontology learning. In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 173–190. Springer.

Maynard, D., Peters, W., and Li, Y. (2006). Metrics for evaluation of ontology-based information extraction. In *Proceeding of the EON 2006 Workshop*.

Mika, P. (2004). Social networks and the semantic web. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '04, pages 285–291, Washington, DC, USA. IEEE Computer Society.

Milch, B., Zettlemoyer, L. S., Kersting, K., Haimes, M., and Kaelbling, L. P. (2008). Lifted probabilistic inference with counting formulas. In *AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence*, pages 1062–1068. AAAI Press.

Miles, A. and Brickley, D. (2005). SKOS core guide. W3C working draft, W3C. Published online on November 2nd, 2005 at `http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102/`.

Muggleton, S. (1996). Stochastic logic programs. In *New Generation Computing*. Academic Press.

Newman, D., Asuncion, A., Smyth, P., and Welling, M. (2007). Distributed inference for latent dirichlet allocation. *Advances in Neural Information Processing Systems*, 20(1081-1088):17–24.

Ng, R. T. and Subrahmanian, V. S. (1990). A semantical framework for supporting subjective and conditional probabilities in deductive databases. Technical report, College Park, MD, USA.

Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *In Proceedings of the 28th International Conference on Machine Learning (ICML 2011)*.

Nixon, L. J. B., Simperl, E., Krummenacher, R., and Martin-Recuerda, F. (2008). Tuplespace-based computing for the semantic web: A survey of the state-of-the-art. *Knowl. Eng. Rev.*, 23:181–212.

Passerini, A., Frasconi, P., and De Raedt, L. (2006). Kernels on prolog proof trees: Statistical learning in the ILP setting. *Journal of Machine Learning Research*, 7:307–342.

Poole, D. (1997). The independent choice logic for modelling multiple agents under uncertainty. *Artif. Intell.*, 94(1-2):7–56.

Poole, D. (2003). First-order probabilistic inference. In *IJCAI'03: Proceedings of the 18th international joint conference on Artificial intelligence*, pages 985–991, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Poon, H. and Domingos, P. (2010). Unsupervised ontology induction from text. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 296–305, Stroudsburg, PA, USA. Association for Computational Linguistics.

Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., and Welling, M. (2008). Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceeding of the 14th ACM SIGKDD international conference*

on Knowledge discovery and data mining, pages 569–577. ACM.

Predoiu, L. (2006). Information integration with bayesian description logic programs. In *Proceedings of the Workshop on Information Integration on the Web (IIWeb 2006), in conjunction with WWW2006*, Edinburgh, Scotland.

Predoiu, L. and Stuckenschmidt, H. (2008). Probabilistic extensions of semantic web languages - a survey. In Ma, Z. and Wang, H., editors, *The Semantic Web for Knowledge and Data Management: Technologies and Practices*, chapter 5. Idea Group Inc.

Punyakanok, V., Roth, D., Yih, W.-t., and Zimak, D. (2005). Learning and inference over constrained output. In *IJCAI'05: Proceedings of the 19th international joint conference on Artificial intelligence*, pages 1124–1129, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on System, Man, and Cybernetics*, 19(1):17–30.

Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. (2010). Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820. ACM.

Rendle, S. and Schmidt-Thieme, L. (2010). Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM 2010: Proceedings of the 2010 ACM International Conference on Web Search and Data Mining*. ACM.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130.

Rettinger, A., Nickles, M., and Tresp, V. (2009). Statistical relational learning with formal ontologies. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, ECML/PKDD2009*, pages 286–301. Springer.

Richardson, M. and Domingos, P. (2006). Markov logic networks. *Journal of Machine Learning Research*, 62(1-2):107–136.

Sato, T., Kameya, Y., and Zhou, N.-F. (2005). Generative modeling with failure in prism. In *IJCAI'05: Proceedings of the 19th international joint conference on Artificial intelligence*, pages 847–852, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Sebag, M. (1997). Distance induction in first order logic. In Džeroski, S. and Lavrač, N., editors, *Proceedings of the 7th International Workshop on Inductive Logic Programming, ILP97*, volume 1297 of *LNAI*, pages 264–272. Springer.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Shervashidze, N. and Borgwardt, K. (2009). Fast subtree kernels on graphs. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A., editors, *Advances in Neural Information Processing Systems (NIPS 2009)*, pages 1660–1668. Neural Information Processing Systems Foundation.

Singla, P. and Domingos, P. (2006). Entity resolution with markov logic. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pages 572–582, Washington, DC, USA. IEEE Computer Society.

Stumme, G., Hotho, A., and Berendt, B. (2006). Towards semantic web mining. *Journal of Web Semantics*, 4(2):124–143.

Takacs, G., Pilaszy, I., Nemeth, B., and Tikk, D. (2007). On the gravity recommendation system. In *Proceedings of KDD Cup and Workshop 2007*.

Taskar, B., Abbeel, P., and Koller, D. (2002). Discriminative probabilistic models for relational data. In *Uncertainty in Artificial Intelligence (UAI)*.

Thor, A., Anderson, P., Raschid, L., Navlakha, S., Saha, B., Khuller, S., and Zhang, X. (2011). Link prediction for annotation graphs using graph summarization. pages 714–729. Springer.

Tiropanis, T., Davis, H., Millard, D., and Weal, M. (2009). Semantic technologies for learning and teaching in the web 2.0 era: A survey of uk higher education. In *Web Science 2009 Conference*.

Tresp, V., Bundschus, M., Rettinger, A., and Huang, Y. (2008). Towards machine learning on the semantic web. In da Costa, P. et al., editors, *Uncertainty Reasoning for the Semantic Web I*, volume 5327 of *LNAI*. Springer.

Tresp, V., Huang, Y., Jiang, X., and Rettinger, A. (2011). Graphical models for relations-modeling relational context. In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval (KDIR)*.

Velardi, P., Navigli, R., Cucchiarelli, A., and Neri, F. (2005). Evaluation of ontolearn, a methodology for automatic learning of ontologies. In Buitelaar, P., Cimmiano, P., and Magnini, B., editors, *Ontology Learning from Text: Methods, Evaluation and Applications*. IOS Press.

Wermser, H., Rettinger, A., and Tresp, V. (2011). Modeling and learning context-aware recommendation scenarios using tensor decomposition. In *Proc. of the International Conference on Advances in Social*

Networks Analysis and Mining.

Xu, Z., Tresp, V., Yu, S., Yu, K., and Kriegel, H.-P. (2007). Fast inference in infinite hidden relational models. In Frasconi, P., Kersting, K., and Tsuda, K., editors, *Proceedings of Mining and Learning with Graphs, MLG2007*.

Yu, K., Chu, W., Yu, S., Tresp, V., and Xu, Z. (2006). Stochastic relational models for discriminative link prediction. In *Advances in Neural Information Processing Systems 19*.

Yu, S., Yu, K., and Tresp, V. (2005). Soft clustering on graphs. In *Advances in Neural Information Processing Systems 18*.