Adaptive Knowledge Propagation in Web Ontologies

PASQUALE MINERVINI, Department of Computer Science - University of Bari, Italy
VOLKER TRESP, Siemens AG, Corporate Technology, Munich, Germany
CLAUDIA D'AMATO, Department of Computer Science - University of Bari, Italy
NICOLA FANIZZI, Department of Computer Science - University of Bari, Italy

We focus on the problem of predicting missing assertions in Web Ontologies. We start from the assumption that individual resources that are *similar* in some aspects are more likely to be linked by specific relations: this phenomenon is also referred to as *homophily*, and emerges in a variety of relational domains. In this article, we propose a method for: (i) *Identifying* which relations in the Ontology are more likely to link similar individuals, and (ii) Efficiently *propagating* knowledge across chains of similar individuals. By enforcing *sparsity* in the model parameters, the proposed method is able of selecting only the most relevant relations for a given prediction task. Our experimental evaluation demonstrates the effectiveness of the proposed method in comparison with state-of-the-art methods from the literature.

General Terms: Ontologies, Semi-Supervised Learning, Transductive Learning, Web of Data

1. INTRODUCTION

In the perspective of the *Semantic Web* (henceforth SW) [Berners-Lee et al. 2001] as a *Web of Data*, standard knowledge representation formalisms are being adopted for publishing data that are semantically annotated along with shared vocabularies (*Web ontologies*). In particular, this phenomenon is testified by the popularity of the *Linked Data* (LD) initiative [Bizer et al. 2009a; Heath and Bizer 2011] and the growth of the *Linking Open Data* (LOD) cloud[1], a set of interlinked datasets, which includes large scale and popular knowledge bases (KBs) such as DBpedia [Auer et al. 2007], Freebase [Bollacker et al. 2008] and YAGO [Suchanek et al. 2007].

Owing to their inherent distributed and dynamic nature and scale, these KBs are often far to be complete. For instance, as of October 2013, 75% of the *persons* in Freebase were missing the *nationality* property value, and coverage for less common properties can be even lower (as discussed also in [Dong et al. 2014]).

In this article we focus on the problem of predicting missing property values of individual resources contained in SW KBs. In the literature this task is often referred to as *assertion prediction* or *knowledge graph completion* [Bordes and Gabrilovich 2014]. Let us consider the following example:

Example 1.1 (*Academic Domain*). Let us consider a KB regarding the academic domain. It contains the following set of assertions for individuals of interest:

$$\begin{aligned}\{ & \texttt{Researcher}(\texttt{MARK}), \texttt{Researcher}(\texttt{LUCAS}), \texttt{Researcher}(\texttt{JOHN}), \\ & \texttt{advisorOf}(\texttt{MARK}, \texttt{LUCAS}), \texttt{worksWith}(\texttt{LUCAS}, \texttt{JOHN}), \\ & \texttt{affiliatedTo}(\texttt{MARK}, \texttt{EFFALG}), \texttt{affiliatedTo}(\texttt{JOHN}, \texttt{COM}) \qquad \}.\end{aligned}$$

Such assertions encode the following facts: (i) Mark, Lucas and John are researchers, (ii) Mark is the advisor of Lucas, and Lucas works with John, and (iii) Mark and John are affiliated, respectively, to the "Efficient Algorithms" (EFFALG) and "Complexity Management" (COM) research groups. Let us assume also that Lucas, as a researcher, has to be affiliated to a research group, but an explicit assertion may not be contained or may not be logically derivable from the KB: it may be one of the previous or some other group. In such cases, one may resort to an *assertion prediction*

[1] As of April 2014, the LOD cloud is composed by 1091 interlinked KBs, describing 8×10^6 entities and 188×10^6 relationships holding between them [Schmachtenberg et al. 2014].

method exploiting available assertions to find the most likely filler x for the assertion $\texttt{affiliatedTo}(\texttt{LUCAS}, x)$. ■

In the literature, this and related problems have often been tackled by exploiting *machine learning* methods [d'Amato et al. 2010; Rettinger et al. 2012; Nayak et al. 2012]. A major issue with existing assertion prediction methods is that they are often computationally impractical, or induce prediction models that are difficult to interpret by domain experts (see Sect. 5 for a detailed discussion on this topic). Our aim is to provide a solution that may cope with such issues.

Contribution. In this article we focus on a method, named *Adaptive Knowledge Propagation* (AKP), for predicting missing property values of individual resources in Web Ontologies. AKP is based on the following intuition: *related entities influence each other*, and those that are linked by specific relations are more likely to share common properties. This phenomenon is referred to as *homophily*, and arises in a vast array of studies on networks [McPherson et al. 2001; Aggarwal 2011]. For example, in social network friends tend to share common characteristics, such as religious beliefs or political views. However, not every relation is equally likely to link entities with similar properties. For instance, it has been observed that talkative persons tend to have silent friends and vice-versa, while partners (in a married couple) are more likely to belong to different genders [Koutra et al. 2011].

Hence, we propose a method for exploiting such heuristics to fill information gaps. In particular, AKP works as follows. (i) First, it *identifies* which relations in the KB are more likely to link similar entities: we refer to such relations as *homophilic relations*. (ii) Then, it leverages such relations for efficiently *propagating* knowledge across chains of related entities. In this way, AKP can effectively predict the value of missing entity properties (such as religious beliefs or political views in a social network domain) in a Web Ontology.

AKP is closely related to Graph-based *Semi-Supervised Learning* (SSL) methods [Chapelle et al. 2006]: such methods rely on a *similarity graph* defined over entities for propagating information across them. The main limitation of Graph-based SSL methods is that they assume that the similarity graph is already given. In this article, we overcome this limitation by proposing a method for learning the *optimal* similarity graph, by leveraging the relationships holding between entities in the KB.

AKP is especially useful with real-world *shallow ontologies* [Shadbolt et al. 2006], which are characterized by a relatively simple terminology and populated by very large amounts of instance data, such as social networks or citation networks. Shallow ontologies like Freebase and YAGO are particularly frequent in the LOD cloud: to organize vast amounts of data, LOD knowledge bases tend to rely on shallow ontologies with low expressiveness and granularity.

Specifically, in this article, we make the following contributions:

- In Sect. 3 we discuss a method, inspired to Graph-based SSL methods, for efficiently *propagating* knowledge among similar instances.
- In Sect. 4 we propose a method for *learning* an optimal similarity graph for a given prediction task: the method leverages a set of semantically diverse relations among examples holding in the ontology.

The method proposed in this article is a significant advance w.r.t. our previous work in [Minervini et al. 2013; Minervini et al. 2012], in which we adopted kernel-defined weights to construct the similarity graph. However, such weights were lacking a meaningful interpretation, as they depend on the topology of the embedding space [Shawe-Taylor and Cristianini 2004] (a common characteristic of many statistical learning models). Moreover, it was observed that they were sensitive to the choice of the hyperparameters.

In this article, we leverage a set of heterogeneous (and possibly complex) relations holding between examples in the ontology for learning an optimal similarity graph. During the construction of such a graph, each relation is associated to a *relevance score*, which has an immediate interpretation in terms of how likely the relation is to link two examples in the same class.

The proposed method is also very efficient: by exploiting recent results on the problem of solving *symmetric and diagonally dominant* linear systems [Cohen et al. 2014; Peng and Spielman 2014], where the coefficient matrices model how information *spreads* across entities, we are able to achieve nearly-linear complexity in both (statistical) inference and learning. For such a reason, the proposed method is scalable and hence suitable for applications involving real, large-scale knowledge bases. We also provide extensive evaluations on the effectiveness of the proposed method in comparison with state-of-the-art methods in the related literature.

Summary. The remainder of this article is organized as follows. In Sect. 2, we review the basics of semantic knowledge representation and reasoning tasks, and we introduce the concept of *transductive learning* in the context of Semantic KBs. Then we illustrate the details of the proposed knowledge propagation method: in Sect. 3 we show a transductive inference procedure using a similarity graph for efficiently propagating knowledge across similar entities so to complete missing values; complementarily, in Sect. 4 we propose a method for learning an optimal similarity graph to be exploited by the propagation procedure. In Sect. 5, we briefly survey related works. In Sect. 6, we experimentally evaluate the proposed method on several datasets. In Sect. 7, we summarize AKP, outline its limitations and discuss possible future research directions.

2. BASICS

In this section, we introduce the basic concepts in this article, including the knowledge representation formalisms and inference services, and the problem of transductive classification in Semantic Web KBs.

2.1. Representation and Deductive Inference

Knowledge Bases. For the sake of generality, we will adopt the notation of *Description Logic* (DL) [Baader et al. 2007] for recalling representation for the KBs and reasoning services to be possibly exploited for retrieving relations between entities (individuals). However, from a more operational viewpoint, the methods investigated in this article can be easily applied to KBs expressed with representation formalisms based on RDF [Hayes and Patel-Schneider 2014] such as OWL2 DL[2] [Grau et al. 2012].

A KB describes a set of objects (or *entities*), their attributes, and the relations between them. The core elements are *atomic concept names* $N_C = \{C, D, \ldots\}$, each interpreted as a subset of objects in the domain (e.g. `Person` and `Article`) and *atomic role names* $N_R = \{R, S, \ldots\}$, each interpreted as a binary relation over the domain (e.g. `friendOf`, `authorOf`). Domain objects (such as persons in a social network, on articles in a citation network) are represented by *individuals* $N_I = \{a, b, \ldots\}$.

Depending on the underlying DL language, a set of constructors is available for building complex concept and role descriptions.

[2] OWL 2 DL is based on $\mathcal{SROIQ}(\mathbf{D})$. We sometimes will use the related terminology with concepts and roles are referred to as *classes* and *properties*, respectively. Classes, properties and individuals are represented by their corresponding IRIs.

Formally, a KB can be seen as made up of three components $\mathcal{K} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$. The *TBox* $\mathcal{T}$ is a set of terminological axioms relating *concepts*, generally *inclusion axioms* ($C \sqsubseteq D$) or *equivalence axioms* ($C \equiv D$). The *RBox* $\mathcal{R}$ is a set of similar terminological axioms that relate *roles*. Finally, the *ABox* $\mathcal{A}$ is a set of extensional axioms, known as *assertions*, relating individuals with concepts and roles, that will be denoted as follows: $C(a)$ (concept assertion) and $R(a, b)$ (role assertion). In the following, we will denote with $\mathsf{Ind}(\mathcal{K})$ the set of individuals occurring in $\mathcal{K}$. The standard DL model theory will be adopted, with $\models$ indicating *logical entailment* with respect to the models of $\mathcal{K}$.

Inference Services. Various inference services are available for querying DL KBs. *Instance Checking* consists in deciding whether $\mathcal{K} \models Q(a)$ or $\mathcal{K} \models r(a, b)$ holds, where Q is a given query concept, and a, b are two individuals. The *Open World Assumption* (OWA) is generally adopted when reasoning over DL KBs and Web ontologies hence it may be not possible to determine the membership of an individual a to some given concept Q and to its complement $\neg Q$, since $\mathcal{K} \models \neg Q(a)$ does not follow from $\mathcal{K} \not\models Q(a)$. This may be caused by an absence of specific disjointness axioms. *Concept Retrieval* is the related inference that aims at collecting individuals that belong to the given query concept: *retrieval*$_\mathcal{K}(Q) = \{a \in \mathsf{Ind}(\mathcal{K}) \mid \mathcal{K} \models Q(a)\}$.

In addition to such inference services, it is also possible to express more complex queries. Given an (infinite) set of variables N_V, a *Conjunctive Query* (CQ) q is a conjunction of concept and role atoms $C(x)$ and $R(x, y)$, with $x, y \in N_V \cup N_I$, built on the signature of $\mathcal{K}$. The set of variables $\mathtt{Var}(q)$ in a conjunctive query q is composed by *answer variables*, and (existentially) *quantified variables*. Informally, a binding of the variables $\mathtt{Var}(q)$ in a CQ q w.r.t. some model of $\mathcal{K}$ determines the *satisfiability* of a query and a result, via the values assigned to the answer variables. Given a CQ q, $\mathcal{K} \models q$ denotes the satisfiability of q w.r.t. all models of the KB $\mathcal{K}$.

2.2. Transductive Learning in Web Ontologies

In this work, we focus on the problem of predicting the missing values of properties for a given set of entities in a KB. To this purpose, we will resort to *transductive learning* [Vapnik 1998] to complement the traditional *inductive learning* setting. While the latter focuses on the creation of general classification models exploiting the available training examples, that can be applied to test instances, the former aims at generalizing directly from training cases, which explicitly have / do not have the given property, to specific test instances. The approach followed in this work, however, can be further generalized to unseen individuals quite simply, using, for example, methods like the one outlined in [Bengio et al. 2006].

Specifically, our method will be able to propagate property information from observed training cases (entities where the considered properties can be observed) to test cases (entities where such properties cannot be either observed, or deductively inferred).

We cast the problem of predicting a missing binary property of individual resources as a *binary classification* problem. The target property may be an explicit concept-membership (i.e. a relation between an individual and a concept) or may be cast as a decision on the membership to the part of a given role domain whose individuals are related to a particular filler. Entities (represented by the individuals occurring in the KB) for which the value of such property is known are considered as *labeled instances*; otherwise, they are considered as *unlabeled instances*.

The aim will be learning a *discriminant function* (also referred to as *labeling function*) defined over the set of examples: given a labeled or unlabeled instance, the function will return a label indicating its class (either positive or negative) that is if the instance has the given property or not.

More formally, the learning problem can be stated in its general form as follows [Chapelle et al. 2006]:

Definition 2.1 (*Transductive Individual Classification*).

Given

- A set of examples $X \subseteq \mathsf{Ind}(\mathcal{K})$, partitioned into:
  - the sets of *positive* and *negative examples*, X_+ and X_-; (*labeled instances*)
  - the set of *neutral examples* X_0 (*unlabeled instances*)

Find A *discriminant function* $\mathrm{f}^* : X \to \{-1, +1\}$, defined over X, assigning one of the two labels, where $+1$ corresponds to the *positive class*, and -1 to the *negative* class.

This labeling function should predict the most likely labels for the unlabeled instances while assigning to the others labels that are coherent with the given classification.

Note that, especially with datasets from the LOD cloud, it may be difficult to find explicit *negative examples* for a given attribute or property. This is often due to the limited expressiveness of the knowledge representation formalism being employed. For instance, a KB expressed in RDF Schema [Guha and Brickley 2014] cannot be inconsistent, except for a few limited cases related to disjoint data-types.

A possible solution to this problem is resorting to a heuristic called the *Local Closed World Assumption* (LCWA) [Galárraga et al. 2013; Dong et al. 2014]: the idea is to consider the knowledge about a specific property R (e.g. `birthDate`) of an individual a to be *locally complete* if a value for R is already specified for the individual a.

More formally, let $O(a, R) = \{b \in \mathsf{Ind}(\mathcal{K}) \mid \mathcal{K} \models R(a, b)\}$ denote the set of individuals in $\mathcal{K}$ related to a by the given role R. Given a candidate assertion $R(a, b)$, its classification (according to the LCWA) is assigned as follows: if $R(a, b) \in O(a, R)$, then a is considered as positive. Conversely, if $R(a, b) \notin O(a, R)$ and $|O(a, R)| > 0$ then a will be assumed as negative, assuming that the $\mathcal{K}$ is *locally complete* for the pair $\langle a, R\rangle$. If $O(a, R) = \emptyset$ then a will be considered as unlabeled (for missing filler on $R(a, \cdot)$), and will have to be predicted by using assertion prediction methods.

This strategy of collecting negative examples by assuming *local completeness* is also adopted in [Lösch et al. 2012; de Vries 2013], two related works on assertion prediction methods. In the empirical evaluations in Sect. 6, for the sake of comparison, we reproduced the experimental settings employed in [Lösch et al. 2012; de Vries 2013] by following the LCWA, and using the unlabeled examples as a source for negative examples under the assumption of local completeness.

Example 2.2 (*Academic Domain – cont.*). Continuing Ex. 1.1, let us consider the task of predicting whether Lucas is affiliated to the *Efficient Algorithms* (`EFFALG`) research group. It can be cast as an *assertion prediction* task, where Mark is a positive example and Lucas an unlabeled example of that research group membership, i.e. $X_+ = \{\texttt{MARK}\}$, $X_0 = \{\texttt{LUCAS}\}$. A problem may arise if, due to the OWA, no *negative example* of the membership to `EFFALG` is available: unless explicitly stated in the KB $\mathcal{K}$, not knowing whether a researcher is in `EFFALG` does not imply that it is not a member of the research group. However, since we know that John is a member of the *Complexity Management* group, one can resort to the LCWA and assume the knowledge about its research group membership is *locally complete*. This allows considering John as a negative example, i.e. $X_- = \{\texttt{JOHN}\}$.

Then, given $X_+ = \{\texttt{MARK}\}$, $X_- = \{\texttt{JOHN}\}$ and $X_0 = \{\texttt{LUCAS}\}$, the research group of Lucas can be recovered by finding a discriminant function $\mathrm{f}^* : X \mapsto \{-1, +1\}$, which associates a binary class (positive or negative) to all researchers, depending on the *predicted* value of their affiliation to the *Efficient Algorithms* research group. ■

It is also possible to use other strategies for collecting negative examples. For instance, (i) one might ask *human experts* to provide for negative examples explicitly, or (ii) add logical axioms to the knowledge base (such as disjointness axioms), so to identify assertions whose truth value is *false*. They may also be learned (e.g. see [Fleischhacker and Völker 2011]).

In Sect. 3, we show how a given *similarity graph*, defined over a set of examples X, can be used for efficiently *propagating* label information from labeled examples in X_+ and X_- to unlabeled examples in X_0, through (chains of) relations in the similarity graph.

In Sect. 4 we show how an optimal similarity graph can be learned from data, by leveraging the relations holding between examples in X in a knowledge base $\mathcal{K}$. In particular, we show that the problem of finding the optimal similarity graph can be cast as an *optimization problem*, which can be solved using gradient-based optimization.

3. KNOWLEDGE PROPAGATION

In this section, we show how a *similarity graph* between examples in X can be used for propagating label information from labeled to unlabeled examples in X.

Let X be a set of $n = |X|$ examples, of which only $l = |X_+ \cup X_-|$, with $l \leq n$, are *labeled* (positive or negative), and the remaining $u = |X_0| = n - l$ are unlabeled (neutral).

We assume we are provided with a weighted undirected *similarity graph* defined over examples X, encoding the similarity relations between examples. Such a graph is represented by its adjacency (weight) matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$, where $\mathbf{W}_{ij}$ is the weight associated to the edge connecting examples $x_i, x_j \in X$. In such a graph, edges with a *strictly positive* weight encode *similarity* relations between examples. If $\mathbf{W}_{ij} > 0$, examples $x_i, x_j \in X$ are linked by a similarity relation with weight $\mathbf{W}_{ij}$. On the other hand, if $\mathbf{W}_{ij} = 0$, there is no edge connecting x_i and x_j.

We can use the similarity graph, represented by $\mathbf{W}$, for propagating label information across similar individuals. Following [Zhu et al. 2003], we can define a *penalty term* (or *cost function*) over labeling functions $\mathbf{f} : X \to \{-1, +1\}$ that penalizes functions $\mathbf{f}$ that do not assign similar labels to examples connected by an edge in the similarity graph. Note that each labeling function $\mathbf{f}$ can also be written as a vector $\mathbf{f} = \left[\mathbf{f}(x_1), \ldots, \mathbf{f}(x_n)\right]^T$, where $\mathbf{f}_i \in \{-1, +1\}$ is the label of the i-th example $x_i \in X$. The penalty term can be defined as follows:

$$\begin{aligned} E(\mathbf{f}) &= \frac{1}{2} \sum_{x_i \in X} \sum_{x_j \in X} \mathbf{W}_{ij} \left[\mathbf{f}(x_i) - \mathbf{f}(x_j)\right]^2 \\ &= \mathbf{f}^T \left(\mathbf{D} - \mathbf{W}\right) \mathbf{f} \\ &= \mathbf{f}^T \mathbf{L} \mathbf{f}, \end{aligned} \tag{1}$$

where $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix such that $\mathbf{D}_{ii} = \sum_{j=1}^{n} \mathbf{W}_{ij}$, and $\mathbf{L} \in \mathbb{R}^{n \times n}$ is the *graph Laplacian* [Spielman 2010], defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$.

Given an input discriminant function $\mathbf{f}$, the penalty term in Eq. (1) associates, for each pair of examples $x_i, x_j \in X$, a non-negative penalty $\mathbf{W}_{ij} \left[\mathbf{f}(x_i) - \mathbf{f}(x_j)\right]^2$. This quantity is 0 when $\mathbf{W}_{ij} = 0$, i.e. when x_i and x_j are not linked in the similarity graph, and when $\mathbf{f}(x_i) = \mathbf{f}(x_j)$. Otherwise, the penalty is strictly positive.

In other terms, the penalty term defined in Eq. (1) encodes our assumption that the *optimal* discriminant function $\mathbf{f}$ should tend to assign the same labels to examples linked by edges with strictly positive weights in the similarity graph. This allows the label information to *propagate* across paths in the similarity graph represented by $\mathbf{W}$.

3.1. Transductive Learning as an Optimization Problem

The penalty term in Eq. (1) penalizes labeling functions that do not assign similar labels to examples connected by an edge in the similarity graph.

Let $L = X_+ \cup X_-$ denote labeled examples, and let $U = X_0$ denote unlabeled examples. The problem of finding an optimal discriminant function $\mathbf{f}^*$ can be cast as an optimization problem, where: (i) $\mathbf{f}^*$ is enforced to be consistent with training labels, and (ii) $\mathbf{f}^*$ minimizes the penalty term $E(\cdot)$, defined in Eq. (1). More formally, the optimal labeling function $\mathbf{f}^*$ can be found by solving the following optimization problem:

$$\begin{aligned} \underset{\mathbf{f} \in \{-1,+1\}^n}{\text{minimize}} \quad & E(\mathbf{f}) \\ \text{subject to} \quad & \forall x \in L : \ \mathbf{f}_i = \mathbf{y}_i, \end{aligned} \tag{2}$$

where $\mathbf{y} \in \{-1, 0, 1\}^n$ is a *label vector* containing labels for labeled examples, such that $\mathbf{y}_i = +1$ (resp. $\mathbf{y}_i = -1$) if $x_i \in X_+$ (resp. $x_i \in X_-$), and $\mathbf{y}_i = 0$ if $x_i \in X_0$.

By minimizing the penalty term $E(\cdot)$ defined in Eq. (1), the optimal labeling function $\mathbf{f}^*$ is likely to assign similar labels to examples connected by an edge in the similarity graph. This allows label information to *propagate* across similar examples.

The constraint $\forall x \in L : \mathbf{f}_i = \mathbf{y}_i$ enforces the labels of all labeled examples $x_i \in L$ to $\mathbf{f}_i = +1$ (resp. $\mathbf{f}_i = -1$) if they are positive (resp. negative) examples: training labels are considered as immutable, under the assumption that they are correct. If training labels can be noisy, it is possible to relax the constraint in Eq. (2) into a *soft constraint*. This relaxation allows the labeling function to relabel training examples, but penalizes the labeling functions that are not consistent with training labels. The new optimization problem can be defined as follows:

$$\underset{\mathbf{f} \in \{-1,+1\}^n}{\text{minimize}} \quad \eta \sum_{x_i \in L} (\mathbf{f}_i - \mathbf{y}_i)^2 + E(\mathbf{f}),$$

where $\eta > 0$ is a user-specified parameter that specifies the magnitude of the penalty associated with inconsistency with training labels. For simplicity, in the following, we assume training labels are correct. Allowing training labels to be noisy does not impact the scalability properties of the proposed method.

However, constraining the discriminant functions $\mathbf{f}$ to only return discrete values (i.e. $\forall x \in X : \mathbf{f}(x) \in \{-1, +1\}$) has two main drawbacks:

(1) Each discriminant function $\mathbf{f}$ can only provide a *hard classification* in $\{-1, +1\}$ (either positive or negative), without yielding any confidence measure.
(2) The penalty term $E(\cdot)$ in Eq. (1) defines the energy function of a discrete Markov Random Field, where calculating the marginal distribution over labels of unlabeled examples is inherently difficult [Koller and Friedman 2009].

For overcoming these problems, in [Zhu et al. 2003] authors propose a continuous relaxation of discriminant function $\mathbf{f}$, by allowing a continuous range of real values as possible outcomes (by using $\mathbf{f} : X \to [-1, +1]$ instead of $\mathbf{f} : X \to \{-1, +1\}$). The relaxation allows defining a much simpler optimization problem, with very interesting complexity properties:

$$\begin{aligned} \underset{\mathbf{f} \in [-1,+1]^n}{\text{minimize}} \quad & E(\mathbf{f}) + \epsilon \mathbf{f}^T \mathbf{I} \mathbf{f} \\ \text{subject to} \quad & \forall x \in L : \ \mathbf{f}_i = \mathbf{y}_i, \end{aligned} \tag{3}$$

where $\epsilon > 0$ is a small weight that (i) guarantees the uniqueness of a global solution to the optimization problem, and (ii) allows the label values to *decay* as the distance from the nearest labeled examples increases in the similarity graph.

Without loss of generality, assume that the vectors $\mathbf{f}$ and $\mathbf{y}$, and the matrices $\mathbf{W}$ and $\mathbf{L}$ are partitioned w.r.t. the membership of examples to the set of labeled examples $L = X_+ \cup X_-$ and unlabeled examples $U = X_0$:

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_L \\ \mathbf{f}_U \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} \mathbf{y}_L \\ \mathbf{y}_U \end{bmatrix}, \qquad \mathbf{W} = \begin{bmatrix} \mathbf{W}_{LL} & \mathbf{W}_{LU} \\ \mathbf{W}_{UL} & \mathbf{W}_{UU} \end{bmatrix}, \qquad \mathbf{L} = \begin{bmatrix} \mathbf{L}_{LL} & \mathbf{L}_{LU} \\ \mathbf{L}_{UL} & \mathbf{L}_{UU} \end{bmatrix}. \tag{4}$$

The optimization problem in Eq. (3) has a unique, global solution, which can be calculated in closed form. The optimal discriminant function is given by $\mathbf{f}^* = \left[\mathbf{f}_L^*, \mathbf{f}_U^*\right]^T$, where $\mathbf{f}_L^* = \mathbf{y}_L$, i.e. the labels for labeled examples in L coincide with training labels, and $\mathbf{f}_U^*$ is calculated as follows:

$$\mathbf{f}_U^* = (\mathbf{L}_{UU} + \epsilon\mathbf{I})^{-1}\mathbf{W}_{UL}\mathbf{y}_L. \tag{5}$$

where $\mathbf{f}_i^* = \mathbf{f}^*(x_i) \in [-1, +1]$ is predicted label for the i-th example $x_i \in X$.

It is important to note that, given an example $x \in X$, $\mathbf{f}^*(x) \approx 1$ (resp. $\mathbf{f}^*(x) \approx -1$) means a high confidence that the example is in the positive (resp. negative) class, while $\mathbf{f}^*(x) \approx 0$ denotes a very low confidence in the labeling, which is given by $\text{sgn}(\mathbf{f}^*(x))$.

An intuitive interpretation for the proposed model is the following: the similarity graph $\mathbf{W}$ can be interpreted as an *electric network* [Bengio et al. 2006] with conductance $\mathbf{W}_{ij}$ between nodes i and j: positive examples are connected to a positive voltage source ($+1V$), negative examples are connected to a negative source ($-1V$). Eq. (5) is a solution to the problem of computing the voltage on the *unlabeled* examples, to assess whether it is positive or negative.

Complexity of Computing the Closed Form Solution. Indeed computing $\mathbf{f}_U^*$ in Eq. (5) can be reduced to solving a linear system in the form $\mathbf{A}\mathbf{x} = \mathbf{b}$, with $\mathbf{A} = (\mathbf{L}_{UU} + \epsilon\mathbf{I})$, $\mathbf{b} = \mathbf{W}_{UL}\mathbf{f}_L^*$ and $\mathbf{x} = \mathbf{f}_U^*$. A linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with $\mathbf{A} \in \mathbb{R}^{n \times n}$ can be solved in nearly linear time if the coefficient matrix $\mathbf{A}$ is SDD. In Eq. (5), the matrix $(\mathbf{L}_{UU} + \epsilon\mathbf{I})$ is SDD since the graph Laplacian $\mathbf{L}$ is SDD [Spielman 2010].

In [Cohen et al. 2014], authors propose an efficient algorithm for solving SDD linear systems: it has an approx. $\mathrm{O}\big(m \log^{1/2} n\big)$ time complexity, where m is the number of non-zero entries in $\mathbf{A}$. An efficient parallel solver for SDD linear systems is also discussed in [Peng and Spielman 2014].

4. LEARNING TO PROPAGATE KNOWLEDGE IN WEB ONTOLOGIES

In Sect. 3, we showed how a *similarity graph* defined over a set of examples X can be used for efficiently propagating label information to all examples in X. Specifically, we showed how this is equivalent to finding an optimal labeling function $\mathbf{f}^* : X \mapsto [-1, +1]$ with respects a given set of properties: *consistency* with training labels, and *smoothness* on the similarity graph. In this section we discuss how we can exploit the relations holding between examples X in the KB for learning an optimal similarity graph.

As already mentioned in Sect. 1, the underlying assumption in this work is that *related individuals influence each other*: some relations may be *homophilic*, in the sense that related individuals may tend to share a set of common properties.

Homophily is the tendency of individuals to associate with similar others [Aggarwal 2011], and it is a phenomenon that occurs in a wide variety of networked domains. For example, in social networks, friends are more likely to share several characteristics such as their political views, occupations, interests and beliefs [Aggarwal 2011; McPherson et al. 2001]. The main problem we tackle in this article is that *this is not always true*: e.g. in social networks, talkative people tend to befriend silent ones and vice-versa [Koutra et al. 2011]. For such a reason, we need to identify which relations are homophilic, before relying on them for propagating knowledge across examples.

Combining Multiple Similarity Graphs. For identifying homophilic relations, the proposed method proceeds as follows. For each relation type $\texttt{rel}_i$ between examples in X (such as `friendOf` or `advisorOf`), we create a corresponding undirected *similarity graph*, represented by the adjacency matrix $\mathbf{R}_i \in \{0, 1\}^{n \times n}$. The matrix $\mathbf{R}_i$ is structured as follows: $\mathbf{R}_{ij} = 1$ iff $\mathcal{K} \models \texttt{rel}(x_i, x_j)$ or $\mathcal{K} \models \texttt{rel}(x_j, x_i)$. While the relation type $\texttt{rel}_i$ may be directed, the corresponding similarity graph $\mathbf{R}_i$ is undirected, since its role is modeling the similarity relationships between examples in X.

Following our intuition that *not every relation is homophilic*, we propose the following model: we represent the adjacency matrix $\mathbf{W}$ of the similarity graph as a (weighted) *linear combination* of the matrices $\mathcal{R} = \{\mathbf{R}_1, \dots, \mathbf{R}_r\}$, where $\mathbf{R}_i$ is associated to the i-th relation type between individuals in X. More formally, we propose the following parametrization for the adjacency matrix of the similarity graph $\mathbf{W}$:

$$\mathbf{W} = \sum_{i=1}^{r} \mu_i \mathbf{R}_i, \qquad \text{with } \mu_i \in \mathbb{R}_+, \forall i \tag{6}$$

where each $\mu_i \geq 0$ is a parameter representing the *weight* of the relational matrix $\mathbf{R}_i$ in the construction of the similarity graph $\mathbf{W}$.

Example 4.1 (*Academic Domain – cont.*). Consider the example in Ex. 2.2, where we casted the problem of predicting missing memberships to the *Efficient Algorithms* research group as an *assertion prediction* problem.

Assume we successfully identified that the advisor/advisee relationships between researchers are *homophilic*, i.e. an advisor and its advisee are likely to be in the same research group. We can construct a similarity graph between examples in X with adjacency matrix $\mathbf{W}$, such that $\mathbf{W}_{ij} = 1$ and $\mathbf{W}_{ji} = 1$ iff $\mathcal{K} \models \texttt{advisorOf}(x_i, x_j)$, with $x_i, x_j \in X$. Note that despite the `advisorOf` relationship is *directional*, the corresponding similarity graph is *undirected*. This is because $\mathbf{W}$ is a model of the (symmetric) similarity relationships between researchers. Given the similarity graph $\mathbf{W}$, encoding our knowledge that advisors are likely to be in the same research groups as their advisees, we can use $\mathbf{W}$ to *propagate* knowledge about research group affiliations across the researchers in X, and successfully recognize that Lucas is a member of the `EFFALG` research group, since his advisor Mark also is. ■

The parameters $\boldsymbol{\mu} = \{\mu_1, \dots, \mu_r\}$ can be either provided by an expert, which already knows which relations are homophilic w.r.t. the properties of interest. As an alternative, parameters $\boldsymbol{\mu}$ can be *learned from data*, as we show in the following.

Parameters Learning

The similarity graph $\mathbf{W}$ is fully specified by the set of parameters $\boldsymbol{\mu}$ in Eq. (6), which may not be known in advance. The parameters $\boldsymbol{\Theta} = \{\boldsymbol{\mu}, \epsilon\}$ fully specify how knowledge *propagates* across the relations between examples in X. In this section, we discuss how the optimal parameters $\boldsymbol{\Theta}$ can be learned from data.

In a *model selection* setting [Bishop 2006], parameters $\boldsymbol{\Theta}$ can be estimated by minimizing a *k-fold Cross Validation* (CV) *Error*. More formally, let the set of labeled examples L be partitioned into k *folds*, and denote as L_i the i-th fold of L, and all other folds as L_{-i}. The k-fold CV Error is defined as the summation of the reconstruction errors obtained by considering the examples in each fold L_i as unlabeled, and predicting their labels. A special case of the k-fold CV Error is the *Leave-One-Out* (LOO) *Error* [Bishop 2006], where $k = |L|$. In the following, we will minimize this quantity, since it overcomes the non-determinism introduced by randomly sampling the k folds.

Formally, let $U_i = U \cup \{x_i\}$ and $L_i = L \setminus \{x_i\}$ be the new sets of labeled and unlabeled examples, obtained by considering the example $x_i \in L$ as *unlabeled*. For simplicity, we assume that x_i is the first element in the enumeration of the set U_i. Furthermore, let $\ell(z, \hat{z})$ be a generic, differentiable loss function which measures the *reconstruction error* between the real label z and the predicted label $\hat{z}$: possible choices for the loss function ℓ are the absolute loss $\ell(x, \hat{x}) = |x - \hat{x}|$ (used in this article), or the quadratic loss $\ell(x, \hat{x}) = (x - \hat{x})^2/2$. The Leave-One-Out Error can be defined as follows:

$$\mathcal{L}(\boldsymbol{\Theta}) = \sum_{i=1}^{|L|} \ell(\mathbf{y}_i, \hat{\mathbf{f}}_i), \tag{7}$$

where $\mathbf{y}_i$ represents the *real* label of example $x_i \in L$, and $\hat{\mathbf{f}}_i$ represents its predicted label, computed assuming that x_i was unlabeled.

The value of $\hat{\mathbf{f}}_i$ can be computed in closed form, by propagating label information from examples in L_i to unlabeled examples in U_i:

$$\hat{\mathbf{f}}_i = \mathbf{e}^T \mathbf{f}^*_{U_i} = \mathbf{e}^T (\mathbf{L}_{U_i U_i} + \epsilon \mathbf{I})^{-1} \mathbf{W}_{U_i L_i} \mathbf{f}_{L_i},$$

where $\mathbf{e} = [1, 0, \ldots, 0]^T$ is a vector used for selecting the first element of the vector of labels inferred by the propagation process, using the closed-form solution in Eq. (5).

The parameters that minimize the LOO Error can be calculated by solving the following constrained optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\Theta}}{\text{minimize}} \quad & \mathcal{L}(\boldsymbol{\Theta}) + \lambda_1 ||\boldsymbol{\Theta}||_1 + \frac{\lambda_2}{2} ||\boldsymbol{\Theta}||_2^2 \\ \text{subject to} \quad & \boldsymbol{\mu} \geq \mathbf{0}, \ \epsilon > 0, \end{aligned} \tag{8}$$

where the function $\mathcal{L}$ is the LOO Error defined in Eq. (7), and $\lambda_1, \lambda_2 \geq 0$ weight a L_1 and an L_2 regularization term over $\boldsymbol{\Theta}$, respectively.

The weights λ_1 and λ_2 are particularly useful, since they allow controlling *complexity* of the parameters $\boldsymbol{\Theta}$. In particular, λ_1 weights a *sparsity-inducing regularizer* [Bach et al. 2012], which controls the number of non-zero coefficients in $\boldsymbol{\mu}$. This allows selecting only a limited number of relations for the propagation process, which leads to more efficient models with better generalization performance.

The constrained optimization problem in Eq. (8) can be solved efficiently by using *Gradient-Based Optimization*, where the search direction for the function minimum is defined by the gradient of the LOO Error function $\mathcal{L}$.

The gradient of the LOO Error function $\mathcal{L}$ w.r.t. a parameter $\theta \in \boldsymbol{\Theta}$ is:

$$\frac{\partial \mathcal{L}(\boldsymbol{\Theta})}{\partial \theta} = \sum_{i=1}^{|L|} \frac{\partial \ell(\mathbf{f}_i, \hat{\mathbf{f}}_i)}{\partial \hat{\mathbf{f}}_i} \left(\mathbf{e}^T \mathbf{Z}_i^{-1} \mathbf{z}_i \right), \quad \text{with } \mathbf{z}_i = \left(\frac{\partial \mathbf{W}_{U_i L_i}}{\partial \theta} \mathbf{f}_{L_i} - \frac{\partial \mathbf{Z}_i}{\partial \theta} \mathbf{f}^*_{U_i} \right). \tag{9}$$

The gradient in Eq. (9) follows from the chain rule and from the gradient of $\hat{\mathbf{f}}_i$:

$$\begin{aligned} \frac{\partial \hat{\mathbf{f}}_i}{\partial \theta} &= \frac{\partial}{\partial \theta} \left(\mathbf{e}^T \mathbf{Z}_i^{-1} \mathbf{W}_{U_i L_i} \mathbf{f}_{L_i} \right) \\ &= \mathbf{e}^T \mathbf{Z}_i^{-1} \left(\frac{\partial \mathbf{W}_{U_i L_i}}{\partial \theta} \mathbf{f}_{L_i} - \frac{\partial \mathbf{Z}_i}{\partial \theta} \mathbf{Z}_i^{-1} \mathbf{W}_{U_i L_i} \mathbf{f}_{L_i} \right) \\ &= \mathbf{e}^T \mathbf{Z}_i^{-1} \left(\frac{\partial \mathbf{W}_{U_i L_i}}{\partial \theta} \mathbf{f}_{L_i} - \frac{\partial \mathbf{Z}_i}{\partial \theta} \mathbf{f}^*_{U_i} \right), \end{aligned}$$

using the properties $\partial(\mathbf{X}^{-1}) = -\mathbf{X}^{-1}(\partial \mathbf{X})\mathbf{X}^{-1}$ and $\partial(\mathbf{X}\mathbf{Y}) = \mathbf{X}(\partial \mathbf{Y}) + (\partial \mathbf{X})\mathbf{Y}$.

A simple gradient-based optimization algorithm, based on *gradient descent*, is outlined in Alg. 1. The algorithm starts by randomly initializing the parameters $\boldsymbol{\Theta}$. Then,

ALGORITHM 1: Projected Gradient Descent for Minimum LOO Error Parameters Learning

```
Input: Training Labels $\mathbf{y}_L$, Threshold $\gamma$, Number of iterations $\tau$:
Output: Minimum LOO Error Parameters $\mathbf{\Theta}^*_{LOO}$.
// Randomly initialize parameters:
$\mathbf{\Theta}^{(0)} \leftarrow Init()$
for $t = 1, \ldots, \tau$ do
    // Gradient descent step
    $\mathbf{\Theta}^{(t)} \leftarrow \mathbf{\Theta}^{(t-1)} - \eta_t \nabla \mathcal{L}(\mathbf{\Theta}^{(t-1)})$
    // Enforce the non-negativity constraints on parameters $\mathbf{\Theta}^{(t)}$
    $\forall \mu_i \in \mathbf{\Theta}^{(t)} : \mu_i \leftarrow \max\{\mu_i, 0\},\ \epsilon \leftarrow \max\{\epsilon, \gamma\}$
end
return $\mathbf{\Theta}^{(\tau)}$
```

for finding a set of parameters that minimizes the loss function $\mathcal{L}$, at each iteration, the algorithm takes a step proportional to the *negative* of the gradient of $\mathcal{L}$ w.r.t. parameters $\mathbf{\Theta}$, so to approach a *local minimum* of the function $\mathcal{L}$. At the t-th iteration, the algorithm takes a step proportional to η_t in the direction of the negative of the gradient. The optimal η_t is found through a simple *line search*, by selecting the value which provides the largest decrement in $\mathcal{L}$. The optimization problem in Eq. (8) is subject to a set of constraints, namely $\boldsymbol{\mu} \geq \mathbf{0}$ and $\epsilon > 0$. For such a reason, we employ a variant of the gradient descent algorithm called *projected gradient descent* [Shor et al. 1985]: after each descent step, the parameters are *projected* in the space of valid parameters. In this case, the projection is equivalent to clamping each weight parameter μ_i to 0 if it becomes negative, and the parameter ϵ to $\max\{\epsilon, \gamma\}$, where $\gamma \in \mathbb{R}_+$ is a small non-negative threshold (we empirically select $\gamma = 10^{-6}$).

4.0.1. Complexity of Evaluating and Minimizing the LOO Error. Calculating the LOO Error $\mathcal{L}$ in Eq. (7) requires iterating over labeled examples in L. Specifically, for each labeled example $x_i \in L$, it requires:

(1) Creating two sets of labeled and unlabeled examples $L_i = L \backslash \{x_i\}$ and $U_i = U \cup \{x_i\}$.
(2) Propagating label information from examples in L_i to examples in U_i.
(3) Evaluating the reconstruction error between the real label $\mathbf{y}_i$ of x_i, and its predicted label $\hat{\mathbf{f}}_i$.

It follows that the complexity of evaluating the LOO Error is dominated by the $|L|$ propagation steps, and it is given by $\mathrm{O}\big(|L| m \log^{1/2} n\big)$, where m and n are defined as in the complexity analysis in Sect. 3.

Evaluating the *Gradient* of the LOO Error in Eq. (9), for computing the steepest descent direction, requires iterating over labeled examples in L. For each labeled example $x_i \in L$, calculating the gradient of $\mathcal{L}$ requires (1) propagating the label information from L_i to U_i, (2) computing the gradient of ℓ, and (3) computing $\mathbf{e}^T \mathbf{Z}_i^{-1} \mathbf{z}_i$.

Computing $\mathbf{Z}_i^{-1}$ might not be feasible if $\mathbf{Z}_i$ is large, since matrix inversion has a time complexity of $\approx \mathrm{O}\big(n^{2.3727}\big)$. However, note that $\mathbf{Z}_i = (\mathbf{L}_{U_i U_i} + \epsilon \mathbf{I})$ is SDD: calculating the term $\mathbf{Z}_i^{-1} \mathbf{z}_i$ in Eq. (9) can be again reduced to solving a linear system in the form $\mathbf{A}\mathbf{x} = \mathbf{b}$, with $\mathbf{A} = \mathbf{Z}_i = (\mathbf{L}_{U_i U_i} + \epsilon \mathbf{I})$ and $\mathbf{b} = \mathbf{z}_i$, where the coefficient matrix $\mathbf{A}$ is SDD. As shown in Sect. 3, the complexity of this task is nearly-linear in the number of non-zero coefficients in $\mathbf{A}$.

Since the propagation step and computing $\mathbf{Z}_i^{-1} \mathbf{z}_i$ have the same asymptotic complexity, it follows that evaluating the gradient of the LOO Error $\nabla \mathcal{L}$ is also $\mathrm{O}\big(|L| m \log^{1/2} n\big)$.

4.1. Retrieving Relations between Entities

As mentioned in the introduction of this article (Sect. 1), we rely on the relations holding between examples in the KB for building the similarity graph. Specifically, in Eq. (6) we expressed the adjacency matrix of the similarity graph $\mathbf{W}$ as a *linear combination*, with weights $\boldsymbol{\mu}$, of relational similarity matrices $\mathcal{R} = \{\mathbf{R}_1, \ldots, \mathbf{R}_r\}$, where $\mathbf{R}_i$ corresponds to the i-th relation type holding between examples in X (such as `friendOf` or `advisorOf`).

For expressing the relations holding between examples in the KB, we rely on *Conjunctive Queries* (CQ), as described in Sect. 2. Conjunctive Queries allow representing a wide variety of relations between examples. For instance, the "co-authorship" relation between examples $x_i, x_j \in X$ can be expressed by the following CQ:

$$\exists z. \left(\texttt{authorOf}(x_i, z) \wedge \texttt{authorOf}(x_j, z)\right),$$

where $z \in N_V$ is a *non-distinguished* variable, representing a work co-authored by both x_i and x_j. Similarly, the "co-authorship of an article in the field of Machine Learning, which won the best paper award" can be retrieved by means of the following CQ:

$$\exists z. \left(\texttt{authorOf}(x_i, z) \wedge \texttt{authorOf}(x_j, z) \wedge \texttt{BestPaper}(z) \wedge \texttt{field}(z, \texttt{MachineLearning})\right).$$

The whole space of *possible relations* between examples, expressed by means of CQ, can be too large to be used in practical applications.

Following [Bhagat et al. 2011], we propose capturing two phenomena holding between the entities in a KB:

Homophily. A direct link between entities (such as friendship and supervision) is correlated with those entities being similar.

Co-citation Regularity. Similar entities tend to refer or connect to the same objects (such as co-authorship and co-working).

More complex relationships can also be captured through the Conjunctive Query framework. However, we experimentally found that only relying on *Homophily* and *Co-citation Regularity* leads to state-of-the-art prediction results, while still leading to interpretable models that can be learned efficiently. Thus, we rely on two types of Conjunctive Queries for expressing relations between each pair of examples $x_i, x_j \in X$:

Simple Queries. Queries representing atomic relations, in the form:

$$\texttt{relation}(x_i, x_j), \tag{10}$$

where $\texttt{relation} \in N_R$ is an atomic role.

Symmetric Queries. Queries representing common relationships, in the form:

$$\begin{aligned} &\exists z. \left(\texttt{relation}(x_i, z) \wedge \texttt{relation}(x_j, z)\right), \text{ and} \\ &\exists z. \left(\texttt{relation}(z, x_i) \wedge \texttt{relation}(z, x_j)\right), \end{aligned} \tag{11}$$

where $\texttt{relation} \in N_R$ is an atomic role, and $z \in N_V$ is a non-distinguished variable.

Efficient Retrieval of Relations Expressed Using Conjunctive Queries. As opposed to other SW query languages, Conjunctive Queries are not an officially specified query language: there is no normative syntax, but there is a general agreement regarding their correct formal interpretation [Hitzler et al. 2009]. For retrieving the relations expressed by CQs as those in Eq. (10) and Eq. (11), we would need to write a distinct CQ for each atomic role in the KB: this may not be feasible if there are many atomic roles in the KB. A solution would be relying on a query language that also allows using variables in place of atomic roles.

As a solution, we propose relying on a SPARQL-DL [Sirin and Parsia 2007] reasoner for retrieving relations between examples. This approach has several advantages:

— SPARQL-DL queries generalize Conjunctive Queries by allowing the use of variables in place of atomic roles.
— SPARQL-DL queries share the same syntax as SPARQL queries [Harris and Seaborne 2013], making it straightforward to apply the same method on RDF KB where a SPARQL endpoint is available.

Example 4.2 (*Retrieving Relations Using SPARQL-DL Queries*). Assume we need to retrieve all relations expressed by *Simple Queries*, as in Eq. (10). Such relations can be retrieved by the following simple SPARQL-DL query:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT ?x ?y ?r WHERE {
  ?x ?r ?y .
  ?r a owl:objectProperty .
}
```

In the results of this query, `x` and `y` are mapped to the entities of interest, and `r` is mapped to an atomic role. Similarly, all relations expressed by *Symmetric Queries* can be retrieved by the following two SPARQL-DL queries:

```
SELECT DISTINCT ?x ?y ?r WHERE {
  ?x ?r _:z .
  ?y ?r _:z .
  ?r a owl:objectProperty .
}
```

```
SELECT DISTINCT ?x ?y ?r WHERE {
  _:z ?r ?x .
  _:z ?r ?y .
  ?r a owl:objectProperty .
}
```

Note that the variable `_:z` is a *non-distinguished variable* which does not need to be materialized in the KB (i.e. represented by an individual in an assertion).

Assume that, for efficiency reason, we are only interested in retrieving relationships between individuals in a specific class, such as `Person`. Constraining the type of the two entities involved in the relationships is straightforward, and only requires adding two triple patterns to the SPARQL-DL query, as follows:

```
  ?x a ns:Person .
  ?y a ns:Person .
```

4.2. Summary of the Proposed Method

The method proposed in this article relies on the relationships holding between entities in a KB for propagating knowledge about their properties. It can be seen as composed by a preliminary *learning* phase, where it identifies *homophilic* relations that can be used for propagating information; and a subsequent *inference* phase, where such relations are efficiently used for learning missing properties of individual resources.

Formally, the proposed method proceeds as follows:

(1) Retrieve the relations holding among the examples in X by using SPARQL-DL queries (see Sect. 4.1), and create a set of adjacency matrices $\mathcal{R} = \{\mathbf{R}_1, \ldots, \mathbf{R}_r\}$.
(2) Find the parameters $\mathbf{\Theta} = \{\boldsymbol{\mu}, \epsilon\}$ that minimize the Leave-One-Out Error by using the proposed Gradient Descent algorithm (see Sect. 4).
(3) Use the relations in $\mathcal{R}$ and the learned weights $\boldsymbol{\mu}$ for constructing the *similarity graph* $\mathbf{W}$ (as in Eq. (6)), and use the efficient closed form solution in Sect. 3 for propagating knowledge across chains of similar examples.

5. RELATED WORK

Several methods have been proposed for predicting the truth value of assertion in Web Ontologies. Approaches proposed in literature include kernel methods (e.g. [Bloehdorn and Sure 2007; Lösch et al. 2012; de Vries 2013]), latent factor models such as probabilistic models (e.g. [Domingos et al. 2008; Rettinger et al. 2009]), methods based on tensor and collective matrix factorization (e.g. [Franz et al. 2009; Tresp et al. 2009; Nickel et al. 2011; Drumond et al. 2012; Nickel et al. 2012]) and energy-based methods (e.g. [Socher et al. 2013; Bordes et al. 2013; Bordes et al. 2014]).

In the following, we briefly summarize the pros and cons for each class of methods, with respect to the method proposed in this work.

Kernel Methods

Kernel methods [Shawe-Taylor and Cristianini 2004] are a class of pattern analysis algorithms used for a variety of tasks, such as clustering, classification, regression and ranking. While classical machine learning algorithms require the instances to be provided in the form of a *feature vector*, kernel methods overcome this limitation by only requiring a user-provided *kernel function* that, given two instances, returns a measure of their *similarity*. For such a reason, kernel methods are particularly popular in the analysis of complex structured objects such as trees or graphs, where deriving a corresponding feature vector representation is non-trivial [Gärtner 2009].

Several kernel functions have been proposed for learning from SW knowledge bases, such as the Weisfeiler-Lehman [de Vries 2013] (WL) and the Intersection SubTree [Lösch et al. 2012] (IST) kernels. For assessing the similarity between two individual resources, both the IST and the WL kernels rely on a set of *syntactic features* of the neighborhood of such resources. Specifically, IST counts the number of common intersection subtrees, while WL estimates the number of common isomorphic subgraphs. Other kernel functions have been proposed in [Bloehdorn and Sure 2007; Fanizzi et al. 2012]: however, they rely on a set of user-specified relational features, which might not be known in advance.

Kernel methods can be very efficient and achieve state-of-the-art predictive performance in several assertion prediction tasks. However, kernel methods induce statistical models, such as separating hyperplanes, in the high-dimensional feature space implicitly specified by the kernel function. The kernel function itself usually relies on purely syntactic features of the relational neighborhood of two individual resources. Both the model induced by the kernel method and the features considered by the kernel function may not necessarily have a direct translation in term of domain knowledge, and may be difficult to leverage in real life knowledge bases.

Latent Factor Models

Latent factor models try to find an explanation to the observed facts in a KB by means of a set of *latent factors*, or unobserved variables.

Such models can be based on probability theory (such as the Infinite Hidden Semantic Model proposed in [Rettinger et al. 2009]), on matrix and tensor factorization (such as TripleRank [Franz et al. 2009], SUNS [Tresp et al. 2009] and RESCAL [Nickel et al. 2011]), or on an energy-based framework [LeCun et al. 2006] (such as the Structured Embeddings model [Bordes et al. 2011], the Neural Tensor Network model [Socher et al. 2013] and the Semantic Matching Energy model [Bordes et al. 2014]).

Models in this class have been proposed for a wide range of applications, such as assertion prediction [Nickel et al. 2012; Socher et al. 2013; Bordes et al. 2014], query answering on factorized probabilistic Knowledge Bases [Krompaß et al. 2014] and with incomplete information [Bordes et al. 2013].

A limitation of these models is that, despite their predictive accuracy, it is not possible in general to interpret the latent factors in terms of domain knowledge, since they do not necessarily need to have an interpretable meaning [Miller et al. 2009].

First-Order Probabilistic Logic Models

A variety of methods in Statistical Relational Learning [Getoor and Taskar 2007] try to overcome the issues in terms of model understandability by combining First-Order Logic (FOL) and statistical models. For instance, Markov Logic [Domingos et al. 2008] relies on a set of weighted FOL formulas for creating a Markov Random Field, modeling the interactions between different assertions in a KB. A problem with methods in this class is that they need a possibly very expensive *search process* for finding the optimal set of rules, or features, for a given assertion prediction task.

Mining Heterogeneous Information Networks

Learning from Semantic Web KBs is closely related to the problem of mining *Heterogeneous Information Networks* (HIN) [Sun et al. 2009; Sun and Han 2012b; Sun and Han 2012a]. An HIN is an information network modeling the interactions between a set of entities: an HIN also carries *type information* about both entities and relations. Thanks to their expressiveness, HINs are particularly suited for representing knowledge of general real-world interactions across diverse domains.

The problem of propagating information in Heterogeneous Information Networks has been discussed in literature. In [Ji et al. 2010], authors propose a method for propagating information across multiple types of nodes, assuming there is a single type of relation in the network. It differs from the present work in the following aspects:

(1) In [Ji et al. 2010] authors assume only entities are typed, without considering the multiple, heterogeneous relation types that might occur in relational domains.
(2) In [Ji et al. 2010] parameters are learned using a simple grid search: this not feasible if the space of parameters is high-dimensional.

In [Luo et al. 2014], authors rely on so-called *meta-paths* for representing more complex relations holding between entities in an HIN. The use of "meta-paths" in [Luo et al. 2014] can be considered analogous to the use of Conjunctive Queries in the present article. However, the work in [Luo et al. 2014] differs from the work in this article in the following aspects: (i) They not propose any efficient way of learning the weight of each meta-path. (ii) They do not discuss the problem of efficiently retrieving the relationships encoded by meta-paths. In the present work, we rely on Conjunctive Queries which have a clear and well-defined semantics, and we show that they can be answered efficiently by existing inference services.

6. EMPIRICAL EVALUATION

In this section, we experimentally evaluate the *Adaptive Knowledge Propagation* (AKP) method proposed in this article, and briefly summarized in Sect. 4.2.

In the following experiments, we aim at evaluating the effectiveness of AKP in predicting missing properties of individual resources in Web Ontologies. For the sake of comparison, we reproduced the same experimental settings used in relevant works in related literature on assertion prediction methods.

Sources and datasets for reproducing the empirical evaluations in this article are available on-line, with an open-source license: https://code.google.com/p/akp/. During experiments, we used an open source DL reasoner [3] for answering the SPARQL-DL queries used for retrieving the relations holding among examples in the KB.

[3] Pellet v2.3.1 – http://clarkparsia.com/pellet/

Table I: Ontologies considered in the experiments

Ontology	DL Language	#Axioms	#Individuals	#Properties	#Classes
AIFB PORTAL	$\mathcal{ALEHO}(\mathcal{D})$	268540	44328	285	49
DBPEDIA 3.9 F.	$\mathcal{ALCH}$	78795	16606	132	251
BGS	$\mathcal{ALI}(\mathcal{D})$	825133	87555	154	6

6.1. Ontologies

We considered three real world ontologies: the DBPEDIA 3.9 Ontology [Bizer et al. 2009b], the AIFB PORTAL Ontology [4], and the BRITISH GEOLOGICAL SURVEY (BGS) Ontology [5]. The characteristics of these ontologies are outlined in Tab. I.

— The DBPEDIA [Bizer et al. 2009b] project builds a large, multilingual KB by extracting structured data from Wikipedia and making it available in the LOD cloud; DBPEDIA 3.9, released in September 2013, describes 4.0 million entities.

— The AIFB PORTAL Ontology is based on the SWRC Ontology and on metadata available from the Semantic Portal of the AIFB Institute. It models the key concepts within a research community, including researchers, articles, technical reports, projects and curses. For instance, in the AIFB PORTAL Ontology, ≈ 500 individuals are members of the class `foaf : Person`, and ≈ 2400 individuals are members of the class `foaf : Document`.

— The BRITISH GEOLOGICAL SURVEY (BGS) Ontology is part of an effort held by the British Geological Survey, a partly publicly funded body for earth science, for publishing geological data (e.g. hydro-geological, gravitational and magnetic data) under OpenGeoscience [6]. The BGS Ontology models several types of entities, such as ≈ 11700 named rock units, their lithogenetic types and geological themes.

6.2. Experimental Setting

For each of the ontologies discussed in Sect. 6.1, we consider a different prediction task, where we aim at completing the missing information about a given property of individual resources The properties to be predicted are already fully available in the initial ontologies, which serve as a gold standard.

Following the evaluation protocols in [Lösch et al. 2012; de Vries 2013], for each prediction task, we partially remove the information to be predicted from the ontology, in a k-fold Cross Validation fashion. At each iteration, this creates a set of entities for which the property to be predicted is available (*labeled examples*) and a set of entities for which it is missing (*unlabeled examples*). As noted in [Lösch et al. 2012; de Vries 2013], it is often not possible to extract *negative examples* from a Web Ontology. For instance, in the DBPEDIA 3.9 Ontology, we can extract all US Presidents affiliated with the Democratic Party, but we cannot extract those that are *provably not affiliated* with such a political party. For collecting negative examples, we follow the strategy used in [Lösch et al. 2012; de Vries 2013]: it is based on the Local Closed World Assumption, discussed in Sect. 1, and consists in sampling negative examples from the examples where the property to be predicted is already valued (under the assumption that the knowledge about the considered property is *locally complete*).

In each experiment, we considered the problem of predicting the membership to each of several classes. For each class, we performed a k-fold Cross Validation (CV), with $k = 10$. Due to the large skew in the distribution of existing and missing properties,

[4] Static dump version `V2012-02-21`, retrieved from http://www.aifb.kit.edu/web/Wissensmanagement/Portal

[5] http://data.bgs.ac.uk/, as of March 2014

[6] https://www.bgs.ac.uk/opengeoscience/

we evaluated the results in terms of the Area Under the Precision-Recall Curve (AUC-PR): the AUC-PR has been shown to be a suitable evaluation metric when the number of negative examples sensibly exceeds the number of positive examples [Davis and Goadrich 2006].

For each method we used the same 10-folds partitioning of the dataset. For such a reason, we report statistical significance tests using a paired, non-parametric difference test (Wilcoxon T test). We also report diagrams showing how using a smaller sample of labeled training examples affects results.

Methods Used in Empirical Evaluations. In experiments, we compared AKP (as summarized in Sect. 4.2) with several state-of-the-art assertion prediction methods with different nature. In AKP, the regularization parameter λ_2 was fixed to $\lambda_2 = 10^{-8}$, while the sparsity controlling regularization parameter λ_1 was selected by cross validation: we report the details for each experiment. We learn the similarity graph used by AKP by minimizing the Leave-One-Out Error, as proposed in Sect. 4. In the formulation of the Leave-One-Out Error, we used the absolute loss $\ell(x, \hat{x}) = |x - \hat{x}|$ for measuring the discrepancy between real and predicted labels.

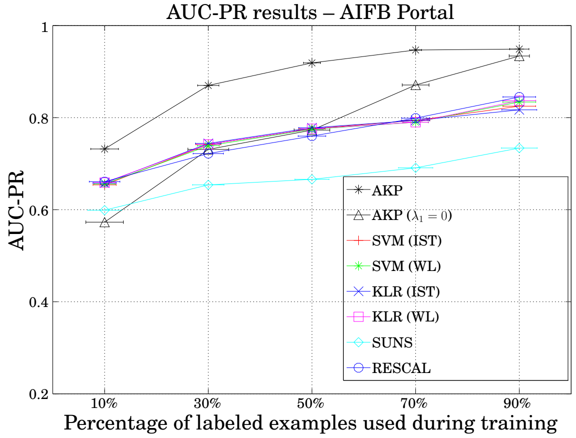
In the comparison, we evaluated two kernel methods: Soft-Margin Support Vector Machine (SVM) [Shawe-Taylor and Cristianini 2004, pg. 223] and Kernel Logistic Regression (KLR) [Hastie et al. 2008]. Each kernel method was used with two different kernel functions aiming at learning from Web Ontologies: the *Intersection Sub-Tree* (IST) kernel [Lösch et al. 2012], and the *Weisfeiler-Lehman* (WL) kernel [de Vries 2013]. As in [Lösch et al. 2012], IST kernel parameters were selected in $d \in \{1, 2, 3, 4\}$ and $\lambda_{ist} \in \{0.1, 0.3, \ldots, 0.9\}$, and WL kernel parameters in $d, h \in \{1, 2, 3, 4\}$ (where d represents the depth of the considered neighborhood graph). The parameter C in SM-SVM was selected in $C \in \{0.0, 10^{-6}, 10^{-4}, \ldots, 10^{4}, 10^{6}\}$, while in KLR the weight λ_k associated to the L_2 regularization term was selected in $\lambda_k \in \{10^{-4}, 10^{-3}, \ldots, 10^{4}\}$.

We also evaluated two latent factor models: SUNS [Tresp et al. 2009] and RESCAL [Nickel et al. 2011]. In the SUNS model, parameters t and λ were selected in $t \in \{2, 4, 6, \ldots, 24\}$ and $\lambda_s \in \{0, 10^{-2}, 10^{-1}, \ldots, 10^{6}\}$. Due to the size of the considered ontologies, in SUNS and RESCAL the RDF graph was composed by nodes corresponding to (labeled and unlabeled) training examples and their neighborhood. The RDF graph used to evaluate kernel functions and latent factor models was materialized as follows: all $\langle \mathtt{s}, \mathtt{p}, \mathtt{o} \rangle$ triples were retrieved by means of SPARQL-DL queries (where $\mathtt{p}$ was either an object or a data-type property) together with all *direct type* and *direct sub-class* relations. For each method, all parameters used in experiments were selected by a k-fold CV within the training set, unless otherwise stated.

6.3. Results

Experiments with the AIFB PORTAL *Ontology.* As in [Lösch et al. 2012; de Vries 2013], the learning task consisted in predicting the affiliations of AIFB staff members to research groups. Specifically, in a set of 316 examples (each representing a researcher in the ontology), the task consisted in predicting missing affiliations to 5 distinct research groups.

The research groups described in the AIFB PORTAL Ontology are *Business Information Systems* (BIK, with 109 affiliates), *Complexity Management* (COM, with 23 affiliates), *Efficient Algorithms* (EFFALG, with 49 affiliates), *Economics and Technology of eOrganizations* (EORG, with 21 affiliates) and *Knowledge Management* (WBS, with 121 affiliates). For each research group, we evaluated the proposed method (jointly with the other methods discussed in Sect. 6.2) on the task of predicting whether unlabeled examples were members of the research group. Following the *Local Closed World Assumption*, discussed in Sect. 2, negative examples for each research group are sam-



Method	AUC-PR (mean ± var.)	
AKP	$\mathbf{0.949 \pm 0.009}$	
AKP ($\lambda_1 = 0$)	0.933 ± 0.010	▽
SUNS	0.734 ± 0.030	▼
RESCAL	0.845 ± 0.025	▼
SMSVM (IST)	0.825 ± 0.025	▼
SMSVM (WL)	0.834 ± 0.025	▼
KLR (IST)	0.817 ± 0.029	▼
KLR (WL)	0.837 ± 0.025	▼

Fig. 1: AIFB PORTAL – Left: AUC-PR results (mean, std.dev.) estimated by 10-fold CV, obtained varying the percentage of labeled examples used for training – Right: AUC-PR results estimated by 10-fold CV: ▼/▽ (resp. ▲/△) indicates that AKP's mean is significantly higher (resp. lower) in a paired Wilcoxon T test with $p < 0.05$ / $p < 0.10$

Table II: AIFB PORTAL – AUC-PR test values on the task of predicting the research group affiliation for all researchers in the AIFB PORTAL Ontology

AIFB	AKP	AKP ($\lambda_1 = 0$)	SVM (WL)	SVM (IST)	SUNS	RESCAL
EFFALG	**.951 ± .051**	.938 ± .063	.838 ± .137	.836 ± .130	.855 ± .098	.890 ± .062
EORG	**.971 ± .092**	.925 ± .163	.956 ± .094	.928 ± .099	.764 ± .173	.842 ± .230
BIK	.905 ± .086	**.921 ± .078**	.824 ± .106	.825 ± .094	.628 ± .131	.887 ± .052
WBS	.972 ± .063	**.993 ± .007**	.875 ± .063	.874 ± .067	.839 ± .080	.809 ± .088
COM	**.944 ± .145**	.885 ± .158	.678 ± .222	.661 ± .226	.586 ± .183	.795 ± .244

pled from the members of other research groups. This procedure is also followed in the experimental evaluations in [Lösch et al. 2012] and [de Vries 2013].

In AKP, the sparsity controlling regularization parameter λ_1 was selected in $\lambda_1 \in \{0, 10^{-8}, 10^{-4}, 10^{-2}, 10^{-1}, 1\}$, according to the performance on a validation set sampled from the training set. For assessing the effectiveness of *sparsity-enforcing regularization*, weighted by λ_1, we also evaluated a variant of AKP, labeled AKP ($\lambda_1 = 0$), where the weight λ_1 was fixed to 0.

In RESCAL, the parameters were selected via 10-fold CV using the training set, with $t \in \{12, 16, \ldots, 32\}$ and $\lambda \in \{10^{-8}, 10^{-4}, 1\}$: due to its computational cost, the number of iterations for the ALS algorithm was fixed to 8, and the graph was composed only by statistical units and their immediate neighborhoods.

Empirical results are described in Fig. 1: the table (right) summarizes the overall AUC-PR results on the research group affiliation prediction task, obtained via 10-fold CV (one per research group, in a *one-versus-all* setting). The plot shows average AUC-PR values describes results obtained with a limited number of labeled training examples, and leaving the rest to the test: error bars represent twice the standard deviation. Detailed results for each research group are available in Tab. II.

From results in Fig. 1, we can clearly see that *AKP yields significantly better AUC-PR results than every other method in the comparison*, where statistical significance was calculated with a Wilcoxon T test with $p < 0.05$.

The main difficulty with this dataset is that, for some researchers, there is very limited information available in the Ontology other than their (to be predicted) affiliation. In these cases, AKP successfully identified it was hardly possible to predict the affiliations of such researchers, and assigned a label $\mathrm{f}_i \approx 0$ to their research group membership, denoting a high degree of uncertainty (recall that $\mathrm{f}_i \in [-1, +1]$).

On the other hand, methods relying on the WL and IST kernels considered researchers with little available information about them similar to each other, and were more likely to assign them to the same research group, even if it is counter-intuitive and not necessarily correct.

The Role of Sparsity. Enforcing sparsity in the parameters, by means of the L_1 regularization term weighted by λ_1, proved to be beneficial for the proposed method. We can clearly see that, on average, AKP yields better results than AKP ($\lambda_1 = 0$), especially when the number of labeled training examples was very limited (i.e. $10\% - 50\%$ of training examples).

A possible explanation for this phenomenon is that AKP ($\lambda_1 = 0$) suffers from the *curse of dimensionality* [Hastie et al. 2008]: since researchers in the AIFB PORTAL Ontology are related by many fine-grained relations, it would require a potentially very large number of labeled training examples for identifying the correct weights $\boldsymbol{\mu}$. For instance, the AIFB PORTAL Ontology contains many highly-specific and fine-grained atomic roles, such as `author2`, `author3`, . . . , `authorN` for indicating that a researcher is the second, third or n-th author of a document. A similar phenomenon happens with the `competenceField` atomic role.

For such a reason, there are many, rather infrequent, relations among researchers, such as “share the same third competence field”, that may happen to relate two researchers in the same research group, but are not always homophilic. For instance, researchers in different research groups may share some non-primary competence fields.

Using a sparsity-enforcing L_1 regularization term sensibly mitigated this problem, by only selecting a limited number of homophilic relations, and leading to simpler, more efficient and more accurate knowledge propagation models.

Qualitative Analysis of Learned Models. AKP correctly identified which relations in the AIFB PORTAL Ontology are more likely to link researchers in the same research group, eliciting new knowledge about this domain.

Tab. III shows a set of the homophilic and non-homophilic relation types discovered during the experiments, from a total of 77 retrieved relation types. Recall that AKP associates each relation type to a weight μ_i, representing its relevancy in the construction of the similarity graph $\mathbf{W}$: relation types with higher μ_i are more likely to be homophilic, and vice versa. For instance, AKP correctly identified that researchers co-authoring the same publications, sharing their main research interests, teaching the same classes and working in the same office are very likely to be affiliated to the same research group.

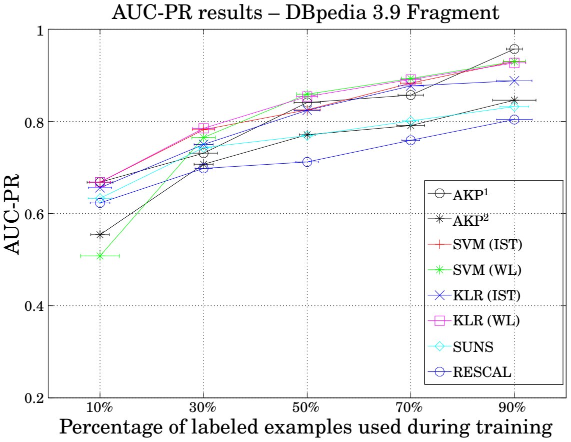
Efficiency. In this experiment, the parameters learning process in AKP took an average of ~ 500 seconds on a single core of an Intel®Core$^{\text{TM}}$i7 processor. This shows that the proposed method is feasible for learning from real world KBs.

6.3.1. Experiments with the DBPEDIA 3.9 *Fragment.* Similarly to [Nickel et al. 2011], we evaluated the proposed approach on the task of predicting political party affiliations to either the Democratic party and the Republican party for 82 US Presidents and Vice-Presidents from the DBPEDIA 3.9 Ontology. The experiment illustrated in [Nickel et al. 2011] uses a small RDF fragment containing the `president` and `vicePresident` predicates only. On the other hand, in this experiment we used a DBPEDIA 3.9 frag-

Table III: Relations between pairs of examples $x_1, x_2 \in X$ considered in the AIFB PORTAL Ontology and the DBPEDIA 3.9 Ontology, and the corresponding weights

AIFB PORTAL	
$\mu_i \gg 0$	$\mu_i \approx 0$
$\exists z.\ [\texttt{publications}(z, x_1) \wedge \texttt{publications}(z, x_2)]$	$\exists z.\ [\texttt{title}(x_1, z) \wedge \texttt{title}(x_2, z)]$
$\exists z.\ [\texttt{interest}(x_1, z) \wedge \texttt{interest}(x_2, z)]$	$\exists z.\ [\texttt{mobile}(x_1, z) \wedge \texttt{mobile}(x_2, z)]$
$\exists z.\ [\texttt{lecturer}(z, x_1) \wedge \texttt{lecturer}(z, x_2)]$	$\exists z.\ [\texttt{road}(x_1, z) \wedge \texttt{road}(x_2, z)]$
$\exists z.\ [\texttt{room}(x_1, z) \wedge \texttt{room}(x_2, z)]$	$\exists z.\ [\texttt{webpage}(x_1, z) \wedge \texttt{webpage}(x_2, z)]$

DBPEDIA 3.9	
$\mu_i \gg 0$	$\mu_i \approx 0$
$\texttt{vicePresident}(x_1, x_2)$	$\texttt{successor}(x_1, x_2)$
$\texttt{president}(x_1, x_2)$	$\texttt{predecessor}(x_1, x_2)$
$\exists z.\ [\texttt{region}(x_1, z) \wedge \texttt{region}(x_2, z)]$	$\exists z.\ [\texttt{profession}(x_1, z) \wedge \texttt{profession}(x_2, z)]$
$\exists z.\ [\texttt{state}(x_1, z) \wedge \texttt{state}(x_2, z)]$	$\exists z.\ [\texttt{award}(x_1, z) \wedge \texttt{award}(x_2, z)]$



Method	AUC-PR (mean ± var.)	S	S+C
AKP^1	**0.957 ± 0.006**		▲
AKP^2	0.846 ± 0.044	▼	
SUNS	0.832 ± 0.019	▼	
RESCAL	0.804 ± 0.029	▼	
SMSVM (IST)	0.930 ± 0.011		▲
SMSVM (WL)	0.930 ± 0.011		▲
KLR (IST)	0.888 ± 0.029	▽	
KLR (WL)	0.927 ± 0.012		△

Fig. 2: DBPEDIA 3.9 Ontology – Left: AUC-PR results (mean, st.d.) estimated by 10-fold CV, obtained varying the percentage of labeled examples used for training – Right: AUC-PR results estimated by 10-fold CV: ▼/▽ (resp. ▲/△) indicates that AKP's mean is significantly higher (resp. lower) in a paired Wilcoxon T test with $p < 0.05$ / $p < 0.10$

ment, obtained through a crawling process, containing a number of potentially irrelevant and possibly noisy entities and relations.

The DBPEDIA 3.9 fragment was extracted through a crawling process, following the extraction procedure proposed in [Hellmann et al. 2009]. Specifically, the RDF graph was traversed starting from resources representing US presidents and vice-presidents: all immediate neighbors were retrieved, together with their related schema information, consisting in direct classes, their super-classes and their subsumption hierarchy. All extracted knowledge was used to create a KB, whose characteristics are summarized in Tab. I.

In AKP, the sparsity controlling regularization parameter λ_1 was selected in $\{0, 10^{-8}, 10^{-4}, 10^{-3}, 10^{-2}, \ldots, 10^{2}\}$ using a 10-fold CV. For efficiency reasons, the number of iterations in the ALS algorithm used by RESCAL was fixed to 16, with parameters $t = 32$ and $\lambda_r = 10^{-8}$ (given by an analysis of the dataset). For the WL kernel, parameters were fixed to $d = 1$ and $h = 1$.

Table IV: DBpedia 3.9 – AUC-PR test values on the task of predicting the political party affiliations for all presidents and vice-presidents in the DBpedia 3.9 Ontology

DBpedia 3.9	AKP^1	KLR (WL)	KLR (IST)	SUNS	RESCAL
Democratic	$\mathbf{.947 \pm .079}$	$.884 \pm .116$	$.879 \pm .102$	$.813 \pm .158$	$.835 \pm .174$
Republican	$\mathbf{.967 \pm .078}$	$\mathbf{.971 \pm 0.091}$	$.897 \pm .224$	$.850 \pm .122$	$.772 \pm .172$

In this experiment, the total number of retrieved relations (both *simple* and *symmetric*) was higher than the number of instances itself: 82 US presidents and vice-presidents were interlinked by 25 *simple* relations and 149 *symmetric* relations. This differs from the other experiments, where instance are only linked by a limited number of, exclusively *symmetric*, relations. For such a reason, we evaluated two variants of the proposed method: AKP^1, which only uses *simple* relations, and AKP^2, which uses both *simple* and *symmetric* relations. Experimental results are summarized in Fig. 2, and AUC-PR results for each distinct political party are outlined in Tab. IV.

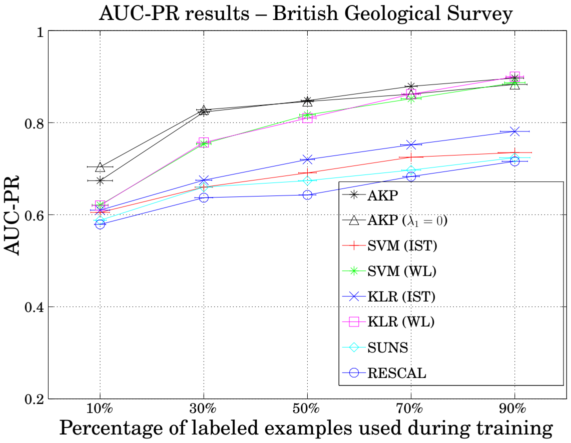
We can see that AKP^1 yields higher AUC-PR values than every other method in the comparison. Specifically, results obtained with AKP^1 were significantly higher than results obtained with AKP^2, SUNS and RESCAL (with $p < 0.05$) and higher than those resulting from KLR with the IST kernel (with $p < 0.1$). This was not true for AKP^2: relying on both simple and symmetric relations greatly increased the variance in AUC-PR results. An explanation is in the *curse of dimensionality*: as the number of considered relations grows, it becomes increasingly difficult to identify those that effectively encode similarities among examples.

Qualitative Analysis of Learned Models. Both AKP^{1}and AKP^{2}successfully identified which relations are likely to link presidents and vice-presidents in the same political party; some of such relations are summarized in Tab. III. The vast majority of such relations was *simple*, suggesting that *homophily* (love of the same) plays a major role on this domain. For instance, both AKP^{1}and AKP^{2}identified that Presidents and their Vice-Presidents, i.e. those linked by `president` and `vicePresident` atomic roles, are very likely to belong to the same political party. In every experiment, the `president` and `vicePresident` relations were assigned the highest weights in every learning task, showing that they can be used for effectively *propagating* political party affiliations.

It was also interesting to note that AKP identified that some *symmetric* relations are homophilic. For instance, AKP^{2}recognized that Presidents and Vice-Presidents coming from the same state, region or district are more likely to be associated with the same political party. It is also remarkable that AKP successfully recognized that not every *simple* relation is homophilic. For instance, AKP recognized that a President and his successor or predecessor (provided by the `successor` and `predecessor` relation, respectively) are unlikely to be members of the same political party. This matches our knowledge that the successor of a Democratic US President is more likely to be Republican, and vice-versa.

Similarly, many *symmetric* relations, even if representing shared characteristics of Presidents and Vice-Presidents, were found not to be relevant with respect to the prediction task at hand. For instance, presidents and vice-presidents sharing their profession, religion or education were not considered more likely to be associated with the same political party.

In AKP^{2}the number of relations considered for constructing the similarity graph was much larger than in AKP^1, and many of such relations were later found irrelevant to the prediction task. This provides a possible explanation to the better results achieved by AKP^{1}in comparison with AKP^2.



Method	AUC-PR (mean ± var.)	
AKP	$\mathbf{0.897 \pm 0.014}$	
AKP ($\lambda_1 = 0$)	0.887 ± 0.012	
SUNS	0.724 ± 0.022	▼
RESCAL	0.716 ± 0.015	▼
SMSVM (IST)	0.735 ± 0.026	▼
SMSVM (WL)	0.887 ± 0.010	
KLR (IST)	0.781 ± 0.020	▼
KLR (WL)	$\mathbf{0.900 \pm 0.007}$	

Fig. 3: BGS Ontology – Left: AUC-PR results (mean, st.d.) estimated by 10-fold CV, obtained varying the percentage of labeled examples used for training – Right: AUC-PR results estimated by 10-fold CV: ▼/▽ (resp. ▲/△) indicates that AKP's mean is significantly higher (resp. lower) in a paired Wilcoxon T test with $p < 0.05$ / $p < 0.10$

Table V: British Geological Survey – AUC-PR test values on the task of predicting the lithogenetic type for all Named Rock Units in the BGS Ontology

BGS	AKP	AKP ($\lambda_1 = 0$)	SM-SVM (WL)	SM-SVM (IST)	SUNS	RESCAL
Fluvial	**.906 ± .075**	**.907 ± .089**	.853 ± .115	.760 ± .146	.703 ± .164	.711 ± .129
Glacial	.889 ± .099	.859 ± .143	**.922 ± .068**	.709 ± .180	.744 ± .133	.720 ± .125

Efficiency. In this experiment, the parameters learning process in AKP took an average of ~ 50 seconds on a single core of an Intel®Core$^{\text{TM}}$i7 processor.

6.3.2. Experiments with the British Geological Survey *Ontology.* As in [de Vries 2013], we evaluated AKP on the *Lithogenesis* prediction problem in the British Geological Survey Ontology. The task consisted in predicting missing lithogenetic information in a set of 159 named rock units. Following [de Vries 2013], we focus on two learning tasks, consisting in the prediction of two major lithogenetic types: "Alluvial" and "Glacial".

In AKP, the sparsity controlling regularization parameter λ_1 was selected in $\lambda_1 \in \{0, 10^{-8}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ using a 10-fold CV. For efficiency reasons, in RESCAL the number of iterations for the ALS algorithm was fixed to 16; parameter selection was performed via 5-fold CV within the training set, with $t \in \{12, 16, \ldots, 32\}$ and $\lambda_r \in \{10^{-8}, 10^{-4}, 1\}$.

Results are summarized in Fig. 3, and grouped for each lithogenetic type in Tab. V. We can see that AKP provides significantly higher AUC-PR values when compared to kernel methods using the IST kernel, SUNS and RESCAL ($p < 0.05$). Also, AKP provides results comparable with those obtained by using the WL kernel, which confirms the effectiveness of the WL kernel on this specific dataset [de Vries 2013]. However, the statistical models produced with the WL kernel can hardly be interpreted in terms of domain knowledge. On the other hand, models learned by AKP explicitly represent the importance of each relation in the knowledge propagation process.

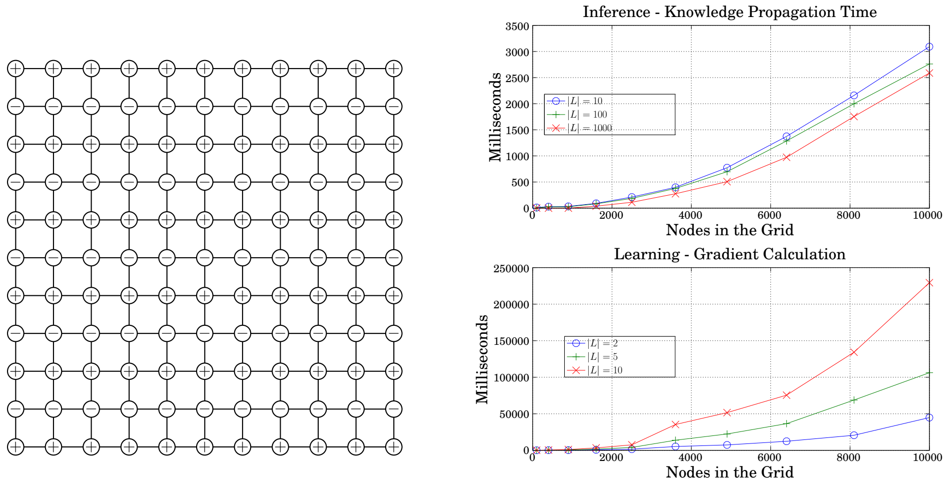


Fig. 4: Left: Grid-structured network – Nodes labeled as + (resp. −) entities in the *positive* (resp. *negative*) class, and only horizontal links represent heterophilic relations. Right: Timings of *knowledge propagation* (inference) and LOO Error *gradient calculation* (learning), for a varying number of nodes in the network and labeled examples.

Also in this case, AKP was able to extract relations between rock units that are likely to link rocks with similar lithogenetic types. For example, among a total of 23 (all *symmetric*) relations, it emerged that rocks with similar geographical distributions, thickness and lithological components were more likely to share their lithogenetic type, while their geological theme and oldest geological age were not considered informative.

6.4. Scalability

As discussed in Sect. 3, in AKP the result of the knowledge propagation process is given by computing the labels for unlabeled examples $\mathbf{f}_U^*$ using the closed-form solution in Eq. (5). Computing $\mathbf{f}_U^*$ is equivalent to solving a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$, with coefficient matrix $\mathbf{A} = \mathbf{L}_{UU} + \epsilon\mathbf{I}$ and $\mathbf{b} = \mathbf{W}_{UL}\mathbf{y}_L$. Since $\mathbf{A} \in \mathbb{R}^{n \times n}$ is SDD, the system can be solved in nearly linear time w.r.t. the number of edges m in the similarity graph, e.g. by using the algorithm proposed in [Cohen et al. 2014] ($\mathrm{O}\left(m \log^{1/2} n\right)$ time complexity). Similar results also apply to the problem of computing the Leave-One-Out Error and its gradient: see the complexity analyses in Sect. 3 and Sect. 4 for more details.

In our experiments, we used an open source implementation [7] of the Lean Algebraic Multigrid (LAMG) [Livne and Brandt 2012], a fast numerical algorithm for solving linear systems with an SDD coefficient matrix. LAMG has a nearly-linear time and space complexity in the number of non-zero elements in the coefficient matrix, and has been shown to scale to graphs up to 47 million edges.

For evaluating the proposed model on relational domains with a growing number of entities, we considered a network structured as the one in Fig. 4 (left): it is structured as a grid, where entities are represented by nodes, and edges represent the relationships between them. Entities on even rows belong to the negative class, while entities

[7] https://code.google.com/p/lamg/

on odd rows belong to the positive class. *Horizontal* and *vertical* relations are of to two distinct relation types, and only the former is *homophilic*.

In Fig. 4 we report the time required for both inference (propagating knowledge across chains of related nodes) and learning (computing the gradient of the Leave-One-Out Error, for finding its steepest descent direction), with a varying number of nodes in the grid and labeled examples. Even for a very large number of nodes (10,000 entities), the closed form solution allows propagating knowledge to the whole graph in less than 3.5 seconds. Computing the gradient of the Leave-One-Out Error is also feasible for very large sets of nodes (10,000 entities). However, since this operation requires a propagation step for each labeled example, its complexity grows with the number of labeled examples $|L|$: this may lead to possibly intractable if the number of labeled examples is very large. A possible solution consists in minimizing the k-fold Cross Validation Error (the Leave-One-Out Error is a special case, with $k = |L|$), discussed in Sect. 4, which would limit the number of required propagation steps to k.

Further Improving the Scalability of the Method. Despite the efficiency of the proposed method and algorithms, it can still be infeasible for very large and Web-scale graphs with Billions of unlabeled examples. Several approaches have been proposed to tackle this problem: they can be used in conjunction with the method proposed in this article for further improving its efficiency and scalability. In [Delalleau et al. 2005], authors propose sub-sampling the examples in the similarity graph, so to reduce the global graph size. In [Bengio et al. 2006], authors propose resorting to an (approximate) iterative propagation process, instead of computing the closed form solution discussed in Sect. 3. In [Zhang et al. 2009], authors propose using the Nyström approximation for representing the Laplacian of the similarity graph. In [Fergus et al. 2009], authors use smooth eigenvectors of the Laplacian of the similarity graph for computing the discriminant function. In [Liu et al. 2010], authors propose using *anchors* (landmarks) for representing groups of nodes in the similarity graph, significantly reducing the size of the graph and thus the complexity of the propagation process. In [Zhang et al. 2011], authors rely on a minimum spanning tree for approximating the similarity graph, and minimum tree cut for propagating information across (chains of) similar examples.

7. CONCLUSIONS AND FUTURE WORK

In this article we proposed a method, named *Adaptive Knowledge Propagation* (AKP) for predicting missing property values for individual resources in Web Ontologies. It relies on the assumption that relations in a knowledge base may be *homophilic* w.r.t. a given property or set of properties, depending on whether they are likely to link *similar* entities. Specifically, in AKP predicting the most likely value for missing properties consists in:

(1) Identifying homophilic relations in the knowledge base, and relying on them for constructing an optimal *similarity graph* (learning phase).
(2) Efficiently propagating knowledge about missing properties of individual resources across the similarity graph (inference phase).

Both phases are discussed in detail, and leverage recent developments in the field of numerical optimization to achieve a nearly-linear time complexity.

We empirically showed that AKP is successful at identifying homophilic relations, and that the extracted knowledge can elicit new knowledge about the domain of interest. We also showed that AKP yields better or very competitive results in comparison with several state-of-the-art assertion prediction methods proposed in literature. Sources and datasets for reproducing the empirical evaluations in this article are available on-line, with an open-source license: https://code.google.com/p/akp/.

REFERENCES

Karl Aberer and others (Eds.). 2007. *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*. Lecture Notes in Computer Science, Vol. 4825. Springer.

Charu C. Aggarwal (Ed.). 2011. *Social Network Data Analytics*. Springer.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A Nucleus for a Web of Open Data, See Aberer et al. [2007], 722–735.

Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider (Eds.). 2007. *The Description Logic Handbook* (2nd ed.). Cambridge University Press.

Francis R. Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. 2012. Optimization with Sparsity-Inducing Penalties. *Foundations and Trends in Machine Learning* 4, 1 (2012), 1–106.

Yoshua Bengio and others (Eds.). 2009. *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada*. Curran Associates, Inc.

Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. 2006. Label Propagation and Quadratic Criterion. In *Semi-Supervised Learning*, Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien (Eds.). MIT Press, 193–216.

Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The Semantic Web. *Scientific American* 284, 5 (May 2001), 34–43.

Smriti Bhagat, Graham Cormode, and S. Muthukrishnan. 2011. Node Classification in Social Networks. See Aggarwal [2011], 115–148.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

Christian Bizer, Tom Heath, and Tim Berners-Lee. 2009a. Linked Data - The Story So Far. *Int. J. Semantic Web Inf. Syst.* 5, 3 (2009), 1–22.

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009b. DBpedia - A crystallization point for the Web of Data. *J. Web Sem.* 7, 3 (2009), 154–165.

Stephan Bloehdorn and York Sure. 2007. Kernel Methods for Mining Instance Data in Ontologies, See Aberer et al. [2007], 58–71.

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, Jason Tsong-Li Wang (Ed.). ACM, 1247–1250.

Antoine Bordes and Evgeniy Gabrilovich. 2014. Constructing and mining web-scale knowledge graphs: KDD 2014 tutorial, See Macskassy et al. [2014], 1967.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014. A semantic matching energy function for learning with multi-relational data - Application to word-sense disambiguation. *Machine Learning* 94, 2 (2014), 233–259.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data, See Burges et al. [2013], 2787–2795.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning Structured Embeddings of Knowledge Bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, Wolfram Burgard et al. (Eds.). AAAI Press.

Christopher J. C. Burges and others (Eds.). 2013. *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*.

O. Chapelle, B. Schölkopf, and A. Zien (Eds.). 2006. *Semi-Supervised Learning*. MIT Press.

Michael B. Cohen, Rasmus Kyng, Gary L. Miller, Jakub W. Pachocki, Richard Peng, Anup Rao, and Shen Chen Xu. 2014. Solving SDD linear systems in nearly $m \log^{1/2} n$ time, See Shmoys [2014], 343–352.

Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. 2010. Inductive learning for the Semantic Web: What does it buy? *Semantic Web* 1, 1-2 (2010), 53–59.

Jesse Davis and Mark Goadrich. 2006. The relationship between Precision-Recall and ROC curves. In *Proceedings of ICML'06*, William Cohen et al. (Eds.). ACM, 233–240.

Gerben Klaas Dirk de Vries. 2013. A Fast Approximation of the Weisfeiler-Lehman Graph Kernel for RDF Data. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD*

2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part I (LNCS), Hendrik Blockeel et al. (Eds.), Vol. 8188. Springer, 606–621.

Olivier Delalleau, Yoshua Bengio, and Nicolas Le Roux. 2005. Efficient Non-Parametric Function Induction in Semi-Supervised Learning. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, AISTATS 2005, Bridgetown, Barbados, January 6-8, 2005*, Robert G. Cowell et al. (Eds.). Society for Artificial Intelligence and Statistics.

Pedro Domingos, Daniel Lowd, Stanley Kok, Hoifung Poon, Matthew Richardson, and Parag Singla. 2008. Just Add Weights: Markov Logic for the Semantic Web. In *Uncertainty Reasoning for the Semantic Web I (LNAI)*, Paulo Cesar G. da Costa et al. (Eds.), Vol. 5327. Springer, 1–25.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion, See Macskassy et al. [2014], 601–610.

Lucas Drumond, Steffen Rendle, and Lars Schmidt-Thieme. 2012. Predicting RDF triples in incomplete knowledge bases with tensor factorization. In *SAC*, Sascha Ossowski et al. (Eds.). ACM, 326–331.

Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. 2012. Induction of robust classifiers for web ontologies through kernel machines. *J. Web Sem.* 11 (2012), 1–13.

Rob Fergus, Yair Weiss, and Antonio Torralba. 2009. Semi-Supervised Learning in Gigantic Image Collections, See Bengio et al. [2009], 522–530.

Daniel Fleischhacker and Johanna Völker. 2011. Inductive Learning of Disjointness Axioms. In *On the Move to Meaningful Internet Systems: OTM 2011 - Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE 2011, Hersonissos, Crete, Greece, October 17-21, 2011, Proceedings, Part II (LNCS)*, Robert Meersman et al. (Eds.), Vol. 7045. Springer, 680–697.

Thomas Franz, Antje Schultz, Sergej Sizov, and Steffen Staab. 2009. TripleRank: Ranking Semantic Web Data by Tensor Decomposition. In *International Semantic Web Conference (LNCS)*, Abraham Bernstein et al. (Eds.), Vol. 5823. Springer, 213–228.

Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. 2013. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, Daniel Schwabe et al. (Eds.). International World Wide Web Conferences Steering Committee / ACM, 413–422.

Thomas Gärtner. 2009. *Kernels For Structured Data*. World Scientific Publishing Co., Inc., River Edge, NJ, USA.

Lise Getoor and Ben Taskar. 2007. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. MIT Press.

Bernardo Cuenca Grau, Peter Patel-Schneider, and Boris Motik. 2012. *OWL 2 Web Ontology Language Direct Semantics (Second Edition)*. W3C recommendation. W3C. http://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/.

Ramanathan Guha and Dan Brickley. 2014. *RDF Schema 1.1*. W3C recommendation. W3C. http://www.w3.org/TR/2014/REC-rdf-schema-20140225/.

Steve Harris and Andy Seaborne. 2013. SPARQL 1.1 Query Language. (March 2013). http://www.w3.org/TR/sparql11-query/

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2008. *The elements of statistical learning: data mining, inference and prediction* (2 ed.). Springer.

Patrick Hayes and Peter Patel-Schneider. 2014. *RDF 1.1 Semantics*. W3C recommendation. W3C. http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/.

Tom Heath and Christian Bizer. 2011. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool.

Sebastian Hellmann, Jens Lehmann, and Sören Auer. 2009. Learning of OWL Class Descriptions on Very Large Knowledge Bases. *Int. J. Semantic Web Inf. Syst.* 5, 2 (2009), 25–48.

Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph. 2009. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC.

Ming Ji, Yizhou Sun, Marina Danilevsky, Jiawei Han, and Jing Gao. 2010. Graph Regularized Transductive Classification on Heterogeneous Information Networks. In *ECML/PKDD (1) (LNCS)*, José L. Balcázar et al. (Eds.), Vol. 6321. Springer, 570–586.

D. Koller and N. Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Danai Koutra, Tai-You Ke, U. Kang, Duen Horng Chau, Hsing-Kuo Kenneth Pao, and Christos Faloutsos. 2011. Unifying Guilt-by-Association Approaches: Theorems and Fast Algorithms. In *Proceedings of ECML/PKDD'11 (LNCS)*, Dimitrios Gunopulos et al. (Eds.), Vol. 6912. Springer, 245–260.

Denis Krompaß, Maximilian Nickel, and Volker Tresp. 2014. Querying Factorized Probabilistic Triple Databases. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part II (LNCS)*, Peter Mika et al. (Eds.), Vol. 8797. Springer, 114–129.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu-Jie Huang. 2006. A Tutorial on Energy-Based Learning. In *Predicting Structured Data*, G. Bakir et al. (Eds.). MIT Press.

Wei Liu, Junfeng He, and Shih-Fu Chang. 2010. Large Graph Construction for Scalable Semi-Supervised Learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, Johannes Fürnkranz et al. (Eds.). Omnipress, 679–686.

Oren E. Livne and Achi Brandt. 2012. Lean Algebraic Multigrid (LAMG): Fast Graph Laplacian Linear Solver. *SIAM J. Scientific Computing* 34, 4 (2012).

Uta Lösch, Stephan Bloehdorn, and Achim Rettinger. 2012. Graph Kernels for RDF Data. In *The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings (LNCS)*, Elena Simperl et al. (Eds.), Vol. 7295. Springer, 134–148.

Chen Luo, Renchu Guan, Zhe Wang, and Chenghua Lin. 2014. HetPathMine: A Novel Transductive Classification Algorithm on Heterogeneous Information Networks. In *Advances in Information Retrieval - 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings (LNCS)*, Maarten de Rijke et al. (Eds.), Vol. 8416. Springer, 210–221.

Sofus A. Macskassy and others (Eds.). 2014. *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*. ACM.

Miller McPherson, Lynn S. Lovin, and James M. Cook. 2001. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology* 27, 1 (2001), 415–444.

Kurt T. Miller, Thomas L. Griffiths, and Michael I. Jordan. 2009. Nonparametric Latent Feature Models for Link Prediction, See Bengio et al. [2009], 1276–1284.

Pasquale Minervini and others. 2012. A Graph Regularization Based Approach to Transductive Class-Membership Prediction. In *Proceedings of the 8th International Workshop on Uncertainty Reasoning for the Semantic Web, URSW2012 (CEUR Workshop Proceedings)*, Fernando Bobillo et al. (Eds.), Vol. 900. CEUR-WS.org, 39–50.

Pasquale Minervini, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. 2013. Transductive Inference for Class-Membership Propagation in Web Ontologies. In *Proceedings of ESWC'13 (LNCS)*, Philipp Cimiano et al. (Eds.), Vol. 7882. Springer, 457–471.

Richi Nayak, Pierre Senellart, Fabian M. Suchanek, and Aparna S. Varde. 2012. Discovering interesting information with advances in web technology. *SIGKDD Explorations* 14, 2 (2012), 63–81.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, Lise Getoor et al. (Eds.). Omnipress, 809–816.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2012. Factorizing YAGO: scalable machine learning for linked data. In *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, Alain Mille et al. (Eds.). ACM, 271–280.

Richard Peng and Daniel A. Spielman. 2014. An efficient parallel solver for SDD linear systems, See Shmoys [2014], 333–342.

Achim Rettinger, Uta Lösch, Volker Tresp, Claudia d'Amato, and Nicola Fanizzi. 2012. Mining the Semantic Web: Statistical Learning for Next Generation Knowledge Bases. *Data Min. Knowl. Discov.* 24, 3 (2012), 613–662.

Achim Rettinger, Matthias Nickles, and Volker Tresp. 2009. Statistical Relational Learning with Formal Ontologies. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part II (LNCS)*, Wray L. Buntine et al. (Eds.), Vol. 5782. Springer, 286–301.

Max Schmachtenberg, Christian Bizer, and Heiko Paulheim. 2014. Adoption of the Linked Data Best Practices in Different Topical Domains. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I (LNCS)*, Peter Mika et al. (Eds.), Vol. 8796. Springer, 245–260.

Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. 2006. The Semantic Web Revisited. *IEEE Intelligent Systems* 21, 3 (2006), 96–101.

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

David B. Shmoys (Ed.). 2014. *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*. ACM.

N. Z. Shor, Krzysztof C. Kiwiel, and Andrzej Ruszcayǹski. 1985. *Minimization Methods for Non-differentiable Functions*. Springer-Verlag New York, Inc., New York, NY, USA.

Evren Sirin and Bijan Parsia. 2007. SPARQL-DL: SPARQL Query for OWL-DL. In *OWLED (CEUR Workshop Proceedings)*, Christine Golbreich et al. (Eds.), Vol. 258. CEUR-WS.org.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning With Neural Tensor Networks for Knowledge Base Completion, See Burges et al. [2013], 926–934.

Daniel A. Spielman. 2010. Algorithms, Graph Theory, and Linear Equations in Laplacian Matrices. In *Proceedings of ICM'10*. 2698–2722.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, Carey L. Williamson et al. (Eds.). ACM, 697–706.

Yizhou Sun and Jiawei Han. 2012a. Mining heterogeneous information networks: a structural analysis approach. *SIGKDD Explorations* 14, 2 (2012), 20–28.

Yizhou Sun and Jiawei Han. 2012b. *Mining Heterogeneous Information Networks: Principles and Methodologies*. Morgan & Claypool Publishers.

Yizhou Sun, Jiawei Han, Peixiang Zhao, Zhijun Yin, Hong Cheng, and Tianyi Wu. 2009. RankClus: integrating clustering with ranking for heterogeneous information network analysis. In *EDBT (ACM International Conference Proceeding Series)*, Martin L. Kersten et al. (Eds.), Vol. 360. ACM, 565–576.

Volker Tresp, Yi Huang, Markus Bundschus, and Achim Rettinger. 2009. Materializing and querying learned knowledge. In *Proceedings of IRMLeS'09*.

Vladimir N. Vapnik. 1998. *Statistical learning theory* (1 ed.). Wiley.

Kai Zhang, James T. Kwok, and Bahram Parvin. 2009. Prototype vector machine for large scale semi-supervised learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009 (ACM International Conference Proceeding Series)*, Andrea Pohoreckyj Danyluk et al. (Eds.), Vol. 382. ACM, 1233–1240.

Yan-Ming Zhang, Kaizhu Huang, and Cheng-Lin Liu. 2011. Fast and Robust Graph-based Transductive Learning via Minimum Tree Cut. In *11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11-14, 2011*, Diane J. Cook et al. (Eds.). IEEE Computer Society, 952–961.

Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. 2003. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In *Proceedings of ICML'03*, Tom Fawcett et al. (Eds.). AAAI Press, 912–919.