Learning Classifiers from Remote RDF Data Stores Augmented with RDFS Subclass Hierarchies

Harris T. Lin*, Ngot Bui†, Vasant Honavar†

*Department of Computer Science
Iowa State University
Ames, IA 50011, USA
htlin@iastate.edu

†Artificial Intelligence Research Laboratory
Center for Big Data Analytics and Discovery Informatics
College of Information Sciences and Technology
Pennsylvania State University
University Park, PA 16802, USA
{npb123, vhonavar}@ist.psu.edu

***Abstract*—Rapid growth of RDF data in the Linked Open Data (LOD) cloud offers unprecedented opportunities for analyzing such data using machine learning algorithms. The massive size and distributed nature of LOD cloud present a challenging machine learning problem where the data can only be accessed *remotely*, i.e. through a query interface such as the SPARQL endpoint of the data store. Existing approaches to learning classifiers from RDF data in such a setting fail to take advantage of RDF schema (RDFS) associated with the data store that asserts subclass hierarchies which provide information that can potentially be exploited by the learner. Against this background, we present a general approach that augments an existing directed graphical model with hidden variables that encode subclass hierarchies via probabilistic constraints. We also present an algorithm ProbAVT that adopts the variational Bayesian expectation maximization approach to efficiently learn parameters in such settings. Our experiments with several synthetic and real world datasets show that: (i) ProbAVT matches or outperforms its counterpart that does not incorporate background knowledge in the form of subclass hierarchies; (ii) ProbAVT remains competitive compared to other state-of-art models that incorporate subclass hierarchies, and is able to scale up to large hierarchies consisting of over tens of thousands of nodes.**

I. Introduction

Resource Description Framework (RDF) offers a formal language for describing structured information on the Web, which emerged as a basic representation format for the Semantic Web over the past decade [1]. Cyganiak [2] estimated in 2011 that there are about 300 interlinked data sets containing over 31 billion triples published in the Linked Open Data cloud covering domains including government, life sciences, geography, social media, and commerce. The increasing availability of large RDF data sets on the web offers unprecedented opportunities for extracting useful knowledge or predictive models from RDF data, and using the resulting models to guide decisions in a broad range of application domains. Indeed, recent effort has considered the use of machine learning approaches, and in particular, statistical relational learning algorithms [3], to extract knowledge from RDF data [4], [5], [6], [7], [8].

However, most existing approaches to learning predictive models from RDF data assume that the learning algorithm has direct access to RDF data. In many settings, it may not be feasible to transfer a massive RDF data set from a remote location for local processing by the learning algorithm. Even in settings where it is feasible to provide the learning algorithm direct access to a local copy of an RDF data set, algorithms that assume in-memory access to data cannot cope with RDF data sets that are too large to fit in memory. Lin et al. [6] presented an approach for constructing Relational Bayesian Classifiers (RBCs) [9] from RDF data using statistical queries through the SPARQL endpoint of the RDF store. More recently, Lin et al. [5] have proposed extensions of this approach for learning a class of generative models from a network of interlinked RDF data stores.

However, RDF triples in an RDF store have often associated with them, RDF Schema (RDFS) [10] that specify a set of classes; these classes organize RDF objects (subjects and objects of predicates) and predicates into type hierarchies as well as domain and range restrictions on RDF predicates (i.e., the type of RDF objects that can appear as subjects or objects of a predicate respectively). RDF schema offer a means to view RDF data at different levels of abstraction. For example, an individual can be described as a *student* at one level of abstraction; or as an *undergraduate* or a *graduate* at a finer level of abstraction; or (in the case of an *undergraduate*) as a *freshman, sophomore, junior or senior.* RDF schema offer the possibility of learning classifiers that are expressed in terms of abstract attribute values leading to simpler, accurate and easier-to-comprehend models that are expressed using familiar hierarchically related attributes. Abstraction provides a form of regularization to minimize overfitting (the finer the level of granularity of description used, the smaller the number of data samples available to estimate the relevant parameters of the predictive models). Current approaches to exploiting abstraction hierarchies to learn from data have serious limitations in settings where the learner does not have direct access to data: (i) Propositionalization techniques that use background knowledge to preprocess the data to obtain a flattened encoding of data [11], [12], [13], [14] using a fixed number of attributes and then use standard supervised learning techniques to build predictive models from such data; (ii) Adaptations of kernel methods such as support vector machines that encode prior knowledge to constrain kernel classifiers [15], [16], [17] which rely on computing a pairwise similarity between data instances (typically in the kernel-induced feature space); (iii) Variants of standard learning algorithms e.g., decision tree learner, naive Bayes learner designed to directly exploit prior knowledge in the form of attribute value taxonomies in a principled fashion to trade off the compactness of the classifiers against their predictive accuracy [18], [19]. However, these approaches assume direct access to data and hence are inapplicable in settings where the data are simply too large to fit in memory.

Against this background, we introduce ProbAVT, an algorithm for learning classifier from a remote RDF data store in settings where the RDF data (ABox in the Semantic Web parlance) is too large to be downloaded across the Internet by the learner and/or to fit in memory, but RDFS (TBox in the Semantic Web parlance) is not. ProbAVT offers a general approach to encode the constraints specified in a subclass hierarchy using hidden (latent or unobserved) variables in a directed graphical model, and adopts the variational Bayesian expectation maximization (VBEM) approach to efficiently learn parameters. Our experiments with several synthetic and real world datasets show that: (i) ProbAVT matches or outperforms its counterpart that does not incorporate background knowledge in the form of subclass hierarchies; (ii) ProbAVT is competitive with other state-of-art models that incorporate subclass hierarchies but assume direct access to data, and is able to scale up to large data sets and large hierarchies.

The rest of the paper is organized as follows: we begin with preliminaries defining and formulating the problem of learning classifiers from RDF data and the associated RDFS subclass hierarchies. We then present a general approach to incorporate subclass hierarchies using hidden variables in a directed graphical model, and adopt the VBEM approach for parameter learning. We use naive Bayes as a demonstrating example to give concrete derivations. Finally we present results of experiments on several synthetic and real world datasets, and conclude with a summary of the key results and a brief discussion of related work.

II. Defining RDF Learners with Subclass Hierarchies

Let I, B, L and V be pairwise disjoint infinite sets denoting the sets of URIs, Blank nodes, Literals and Variables respectively. An RDF triple is of the form $(s, p, o) \in (I \cup B) \times I \times (I \cup B \cup L)$ where s is the subject, p the predicate, and o the object. An RDF graph is a set of RDF triples. Given an RDF graph $\mathcal{G}$, the set of resources is the union of all subjects and objects in $\mathcal{G}$. In this paper, we assume that an RDF graph may include subclass hierarchies, i.e. it includes the following set of reserved predicates: `rdfs:subClassOf` (`sc`), `rdfs:domain` (`dom`), `rdfs:range` (`range`), and `rdf:type` (`type`). Given an RDF graph $\mathcal{G}$, the TBox denoted by $\mathcal{G}_T$ is the subset of $\mathcal{G}$ defined by $\{(s, p, o) \in \mathcal{G} : p \in \{\texttt{sc}, \texttt{dom}, \texttt{range}\}\} \cup \{(s, \texttt{type}, o) \in \mathcal{G} : o \in \{\texttt{rdfs : Class}, \texttt{rdfs : Property}\}\}$. The ABox of $\mathcal{G}$ denoted by $\mathcal{G}_A$ is $\mathcal{G} \setminus \mathcal{G}_T$.

Given an RDF graph $\mathcal{G}$, and a *target class* $\mathcal{T}$ which is a distinguished URI of type `rdfs:Class` in $\mathcal{G}$, we denote the set of instances of the target class as $\mathcal{T}(\mathcal{G}) = \{x : \ (x, \texttt{type}, \mathcal{T}) \in \mathcal{G}\}$. An *attribute* A (of a target class $\mathcal{T}$) is a tuple of predicates $(p_1, \ldots, p_N)$ such that the domain of p_1 is $\mathcal{T}$, the range of p_n is the domain of p_{n+1}, and the range of p_N is a literal. Given an instance x_i of the target class $\mathcal{T}$ and an attribute $A_k = (p_1^k, \ldots, p_J^k)$, we define B_k^i to be the bag (multi-set) of literals matched by the variable $?\,v_J$ in the Basic Graph Pattern [20] $((x, p_1^k, ?\,v_1)$ AND $(?\,v_1, p_2^k, ?\,v_2) \ldots (?\,v_{J-1}, p_J^k, ?\,v_J))$ where $v_j \in V$ are variables. A *target attribute* is a distinguished attribute denoted by A_c, which describes the *class label* of an instance, hence we assume that each instance $x_i \in \mathcal{T}(\mathcal{G})$ has exactly one class label, denoted by c_i, and the set of all possible values of c_i is denoted by $\mathcal{C}$.

An *RDF data set* D is a tuple $(\mathcal{G}, \mathcal{T}, \mathcal{A}, A_c)$ where $\mathcal{G}$ is an RDF graph, $\mathcal{T}$ a target class in $\mathcal{G}$, $\mathcal{A} = (A_1, \ldots, A_K)$ a tuple of attributes, and A_c is a target attribute. Given an RDF data set $D = (\mathcal{G}, \mathcal{T}, \mathcal{A}, A_c)$, its induced *multiset attributed data set* [6] is defined as $\mathcal{M}(D) = \{((B_1^i, \ldots, B_K^i), c_i) : \ x_i \in \mathcal{T}(\mathcal{G})\}$. Let R be an `rdfs:Class`, we denote $Sub(R)$ to be the set of all subclasses of R (including R itself). In this work we further assume that the range of each attribute A_k can be asserted by a subclass hierarchy R_k, and we denote $\mathcal{R} = (R_1, \ldots, R_K)$.

Definition 1. The input to an RDF_{sc}[1] node classifier h is $(B_1^i, \ldots, B_K^i)$ where x_i is an instance of a target class $\mathcal{T}$, and the output $h(x_i) \in \mathcal{C}$ is a class label.

An RDF_{sc} Learner L is an algorithm that given an RDF data set $D = (\mathcal{G}, \mathcal{T}, \mathcal{A}, A_c)$ asserted with a tuple of subclass hierarchies $\mathcal{R}$, its induced multiset attributed data set $\mathcal{M}(D)$, a hypothesis class H, and a performance criterion P, outputs a classifier $h \in H$ that optimizes P.

III. Learning RDF Classifiers with Subclass Hierarchies

We propose to model $\mathcal{M}(D)$ using a generative process that incorporates the abstraction provided by subclass hierarchies. In general, let us assume we are given a bayesian network for $\mathcal{M}(D)$ without the consideration of subclass hierarchies, also assume that an observed variable x can be asserted by a subclass hierarchy R (see Fig. 1(a)). We introduce a hidden (latent) variable a that takes a value from all nodes in the subclass hierarchy R, then encode the subclass hierarchy by

[1] denotes RDF with subclass hierarchies. Further inclusion with subproperty hierarchies would result in an expressivity of RDFS, however it is out of scope of this paper and left as future work.

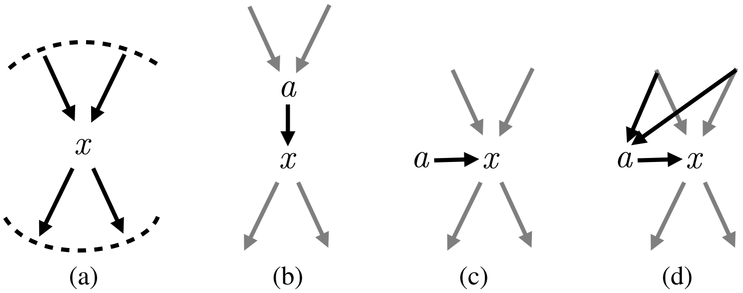

Fig. 1. A graphical model representation of the proposed approach to learn with subclass hierarchies. (a) Original bayesian network where x can be asserted by some subclass hierarchy. (b-d) Examples of how the network can be augmented using the *abstraction* variable a.

placing the constraint: $p(x \mid a) = 0$ if x is not a descendent of a in R. We interpret a as the *abstraction* that represents the observed value x. In general, there are multiple ways to incorporate the hidden variable a into the model (see Fig. 1(b-d) for three examples).

To make our discussion more concrete, we focus on $\mathcal{M}(D)$ using a simple relational naive Bayes of K attributes shown in Fig. 2(a), which has the following joint distribution:

$$p(c, \mathbf{x}) = p(c) \prod_{k=1}^{K} \prod_{n=1}^{N_k} p(x_{kn} \mid c) \tag{1}$$

Now we can apply the three extensions shown in Fig. 1(b-d) to yield corresponding models shown in Fig. 2(b-d). For example, the joint distribution for Fig. 2(b) reads:

$$p(c, \mathbf{a}, \mathbf{x}) = p(c) \prod_{k=1}^{K} \prod_{n=1}^{N_k} p(a_{kn} \mid c)\, p(x_{kn} \mid a_{kn}) \tag{2}$$

Recall that we intend to interpret a_{kn} as the *abstraction* that represents the observed value x_{kn}, hence, alternatively for Eq. 2 we can write $p(a_{kn} \mid c, x_{kn}) \propto p(c)\, p(a_{kn} \mid c)\, p(x_{kn} \mid a_{kn})$ to describe the *distribution* of abstractions that best represent a value x_{kn} given class.

To learn the parameters for the abstraction-augmented graphical models, we take the variational bayesian approach as in [21] where we wish to approximate the log marginal likelihoods given by:

$$\ln p(c, \mathbf{x} \mid m) = \ln \int d\boldsymbol{\theta}\, p(\boldsymbol{\theta} \mid m) \prod_{i=1}^{I} \sum_{\mathbf{a}_i} p(c_i, \mathbf{a}_i, \mathbf{x}_i \mid \boldsymbol{\theta}) \tag{3}$$

We derive the lower bound by applying Jensen's inequality via variational distributions $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ and $\{q_{\mathbf{a}_i}(\mathbf{a}_i)\}_{i=1}^{I}$.

$$\begin{aligned}\ln p(c, \mathbf{x} \mid m) \geq & \int d\boldsymbol{\theta}\, q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \ln \frac{p(\boldsymbol{\theta} \mid m)}{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \\ & + \sum_{i=1}^{I} \int d\boldsymbol{\theta}\, q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \sum_{\mathbf{a}} q_{\mathbf{a}_i}(\mathbf{a}_i) \ln \frac{p(c_i, \mathbf{a}_i, \mathbf{x}_i \mid \boldsymbol{\theta}, m)}{q_{\mathbf{a}_i}(\mathbf{a}_i)}\end{aligned} \tag{4}$$

We will demonstrate the corresponding EM steps using the model specified by Fig. 2(b), where the corresponding steps for the other models can be derived analogously. Let N_{ca} be the expected total number of times variable $A = a$ when its parent $C = c$, similarly let N_{ax} be the expected total number of times variable $X = x$ when its parent $A = a$. We also let N_{cx} be the number of times $C = c$ and $X = x$. The corresponding variational E-step and M-step can be derived as follows [21].

Variational E-step:

$$q_{\mathbf{a}_i}(\mathbf{a}_i) \propto p\left(c_i, \mathbf{a}_i, \mathbf{x}_i \mid \tilde{\boldsymbol{\theta}}\right) \tag{5}$$

Variational M-step:

$$\ln \tilde{\boldsymbol{\theta}}_{ca} = \psi(\lambda_{ca} + N_{ca}) - \psi\left(\sum_{a'} \lambda_{ca'} + N_{ca'}\right) \tag{6}$$

$$\ln \tilde{\boldsymbol{\theta}}_{ax} = \psi(\lambda_{ax} + N_{ax}) - \psi\left(\sum_{x'} \lambda_{ax'} + N_{ax'}\right) \tag{7}$$

A. Obtaining Sufficient Statistics for EM

To construct the model shown in Fig. 2(b), we need to obtain only the relevant counts from data, e.g., N_{cax}, the count of instances where $C = c$, $A = a$, and $X = x$ for the E-steps. In the M-steps, both N_{ca} and N_{ax} can be derived from N_{cax}. In general, the sufficient statistics are equivalent to the statistics needed to learn the original model, and the statistics for bayesian networks for instance can be obtained by querying the RDF data source using an RDF query language such as SPARQL [6].

IV. Experiments

We present three sets of experimental results using datasets with subclass hierarchy information: one using datasets from UCI repository; one using RDFS data; and a synthetic data set with subclass hierarchies of variable size (to assess the scalability of the approach as a function of the size of the hierarchy).

A. UCI datasets with Subclass Hierarchy

We use three datasets (with only discrete attributes) from UCI repository (i.e., Mushroom, Soybean, and Nursery) where the subclass hierarchies are supplied by domain experts. These datasets correspond to a special case for Fig. 2(b-d) where $N_k = 1$ for all k, and the goal of this set of experiments

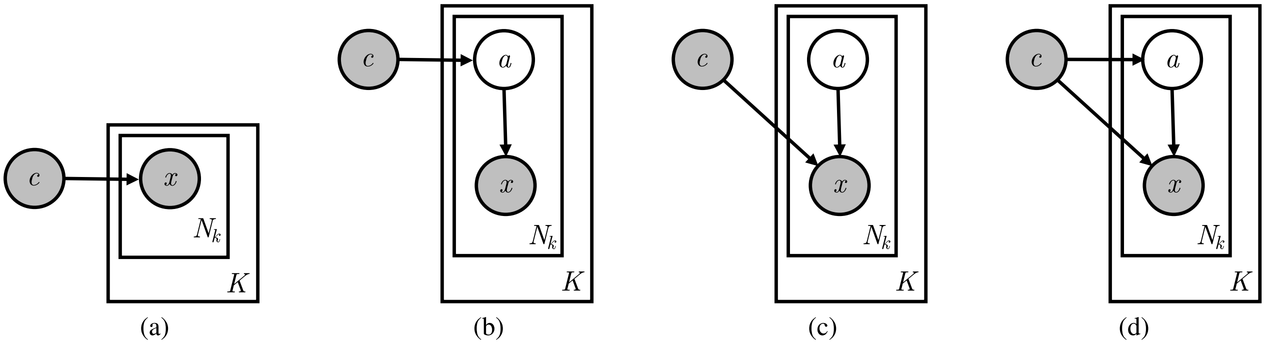

Fig. 2. A naive Bayes instantiation of the proposed model to learn with subclass hierarchies.

Dataset	NB	NB-AVT	ProbAVT(best)	ProbAVT(b)	ProbAVT(c)	ProbAVT(d)
Mushroom	95.83%	*99.85%	99.40%	99.40%	99.14%	99.18%
Soybean	92.09%	*94.73%	*94.29%	93.85%	92.83%	94.29%
Nursery	90.32%	90.32%	90.25%	90.25%	90.19%	90.22%

TABLE I
ACCURACY RESULTS OF THREE UCI DATASETS.

is to compare the proposed method (denoted by ProbAVT) against an alternative model proposed by [19] that extends naive Bayes with subclass hierarchies (denoted by NB-AVT), as well as the naive Bayes without subclass hierarchies as a baseline (denoted by NB).

Table I shows the accuracy results of 10-fold cross validation. Starred numbers represent best performing methods under t-test with $\alpha = 0.05$. We observe that ProbAVT compares favorably against other methods, with the exception of Mushroom dataset, while in Nursery dataset all three methods do not display significant differences. In terms of their run time performance, we observe that both NB and ProbAVT are relatively fast while NB-AVT is significantly slower than the former. We conduct a detailed scalability analysis in Sec. IV-C.

B. RDF datasets with Subclass Hierarchy

1) Web Service Dataset: We use an RDF benchmark dataset from OWLS-TC v2.1[2] service retrieval test collection that contains 578 OWL-S Semantic Web service descriptions. The attributes are input and output concepts described by various subclass hierarchies (and even richer ontologies) consisting over 5000 concepts, and the dataset corresponds to the general case where $N_k \geq 1$, hence we compare our method against two other relational methods: a relational Bayesian classifier [9] (denoted RBC) and a modified relational probability tree [22] that incorporates subclass hierarchies (introduced as part of SPARQL-ML toolkit [8] and hence we denote it by SML). Table II shows the accuracy results of 10-fold cross validation, and that ProbAVT significantly outperforms other methods. In terms of their run time performance, both RBC and ProbAVT completed under 5 minutes (SML run time was not reported in [8]).

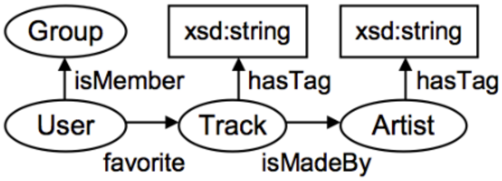

Fig. 3. RDF schema representation of Last.fm dataset.

2) Social Network Dataset: We crawled the Last.fm[3] dataset, a real-world music social network. We manually identified 11 disjoint groups (categories of users who share similar interests in music e.g., http://www.last.fm/group/Metal in the case of users who enjoy Heavy Metal) that contain approximately equal number of users in the network; we then crawled users, items, and the links that denote the relations among the objects in the network, which can be naturally represented as an RDF graph (see Fig. 3 for an example RDF schema representation). In particular, the subset of the Last.fm data that we use consists of 25471 objects. These objects belong to one of 4 types: 10197 users, 8188 tracks, 1651 artists, and 5435 tags. A user listens to a track which is sung by some artist, and users can add tags to tracks as well as artists. We aggregate three attributes for each user: track, artist, and tags. Since there are no natural subclass hierarchies available for this dataset, we used AVT-Learner [23] to build a subclass hierarchy based on class distribution for each attribute, independently for each fold in the cross validation. Our task on this dataset is to classify users where each of them belong to one of 11 categories (groups). Table III shows the accuracy results of 10-fold cross validation. While we observe that ProbAVT significantly outperforms RBC, we also observe significant variation in performance among the three considered ProbAVT models (b-d). In addition all ProbAVT models perform significantly slower than RBC for this dataset (see the next section for further analysis).

[2] http://projects.semwebcentral.org/projects/owls-tc/

[3] http://www.last.fm

RBC	SML	ProbAVT(best)	ProbAVT(b)	ProbAVT(c)	ProbAVT(d)
90.83%	82.88%	*94.29%	94.29%	90.83%	92.21%

TABLE II
ACCURACY RESULTS OF THE WEB SERVICE DATASET.

RBC	ProbAVT(best)	ProbAVT(b)	ProbAVT(c)	ProbAVT(d)
48.57%	*51.33%	46.31%	51.33%	43.65%

TABLE III
ACCURACY RESULTS OF THE LAST.FM DATASET.

C. Scalability Evaluation

We generated a synthetic dataset to evaluate the scalability of our proposed approach, specifically over the size of the subclass hierarchy. We generate balanced binary class datasets with a single attribute. We also generated subclass hierarchies as fully balanced binary trees with levels ranging from 2 to 14, and for each dataset, we randomly assign a probability distribution over the leaves of the hierarchy independently for each class. For each hierarchy size, we then generated 4 datasets consisting of N instances per class where $N \in \{100, 500, 1000, 10000\}$. Finally we generated 5 sets with different random seeds for each dataset configuration, then run NB-AVT and our proposed ProbAVT learner on a machine with 2.8GHz CPU with 16 cores. We report the average of run times for all 5 sets.

Fig. 4 shows the results of this experiment. First we observe that the run time for both methods depends on the size of hierarchy, but both methods scales well when the number of instances is small (100 per class), this is the case even up to hierarchy sizes over 10000 nodes. However, the run times begin to differ significantly when the number of instances increases. When there are 10000 instances per class, we can see that NB-AVT is an order of magnitude slower than ProbAVT. This degradation can be attributed to the NB-AVT method that performs a search over the cut space on the subclass hierarchy, which uses the conditional log likelihood to define the cut refinement criterion that requires computation performed on each individual instance.

On the other hand, when the hierarchy size increases over 10000, we still observe an exponential time increase for ProbAVT especially for larger instance size over 10000. This could be explained by an exponential increase on the number of variational Bayesian E-steps and M-steps taken to converge to a solution. It is of interest to investigate methods to achieve a faster convergence time for VBEM.

We also implemented ML/MAP EM to compute a point estimate of the parameters, and as in [21] we observe that the ML/MAP alternative can easily get trapped in a local minimum depending on the initial conditions. Variational Bayesian EM on the other hand computes a distribution over parameters and naturally incorporates a model complexity penalty, hence offers an explanation of superior and more stable performance.

V. SUMMARY AND DISCUSSION

A. Summary

Rapid growth of RDF data in the Linked Open Data (LOD) cloud offers unprecedented opportunities for analyzing such data using machine learning algorithms. The massive size and distributed nature of LOD cloud present a challenging machine learning problem where the data can only be accessed *remotely*, i.e. through a query interface such as the SPARQL endpoint of the data store. Existing approaches to learning classifiers from RDF data in such a setting fail to take advantage of RDF schema (RDFS) associated with the data store that asserts subclass hierarchies which provide information that can potentially be exploited by the learner. Against this background, we present ProbAVT, an algorithm for learning classifier from RDF data and the associated schema. ProbAVT encodes the constraints specified in a subclass hierarchy using hidden variables in a directed graphical model, and adopts the variational Bayesian EM approach to efficiently learn parameters. Our experiments with several real world datasets show that: (i) ProbAVT matches or outperforms its counterpart that does not incorporate background knowledge in the form of subclass hierarchies; (ii) ProbAVT remains competitive compared to other state-of-art models that incorporate subclass hierarchies, and is able to scale up to large hierarchies over tens of thousands of nodes.

B. Related Work

ProbAVT introduced in this paper extends RBC-Learner [6] to exploit background knowledge in the form of RDFS subclass hierarchies, and yet still learns without direct access to RDF data.

Other approaches to exploiting abstraction hierarchies to learn from data have serious limitations in settings where the learner does not have direct access to data: (i) Propositionalization techniques that use background knowledge to preprocess the data to obtain a flattened encoding of data [11], [12], [13], [14] using a fixed number of attributes and then use standard supervised learning techniques to build predictive models from such data; (ii) Adaptations of kernel methods such as support vector machines that encode prior knowledge to constrain kernel classifiers [15], [16], [17] which rely on computing a pairwise similarity between data instances (typically in the kernel-induced feature space); (iii) Variants of standard learning algorithms e.g., decision tree learner, naive Bayes learner designed to directly exploit prior knowledge in the form of attribute value taxonomies in a principled

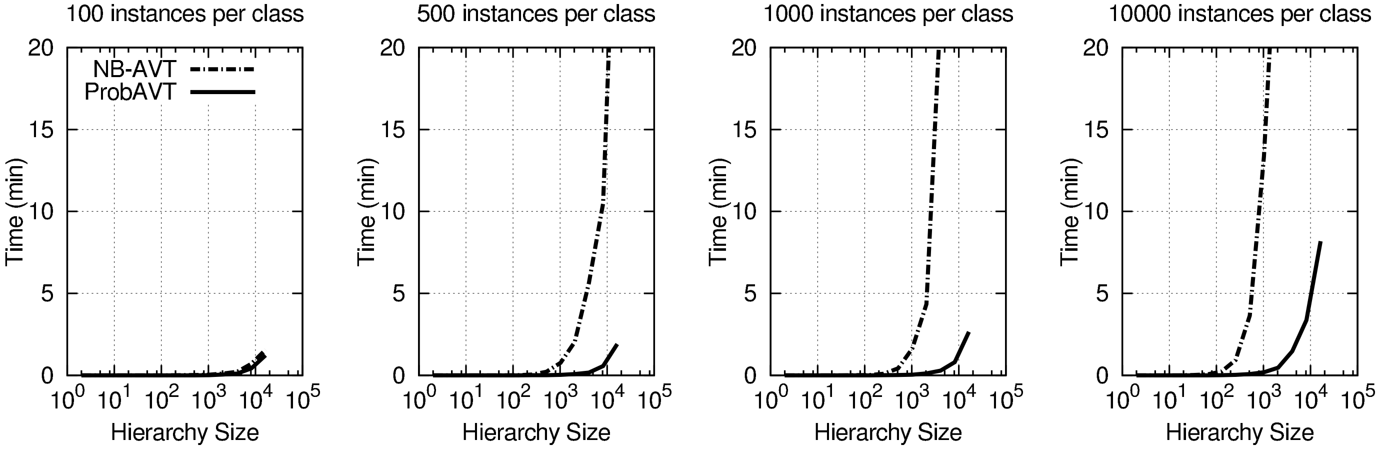

Fig. 4. Run time results with varying hierarchy size and instance size.

fashion to trade off the compactness of the classifiers against their predictive accuracy [18], [19]. However, these approaches assume that the learner has direct access to all RDF data (both TBox and ABox). In contrast, our approach only assumes that the learner has direct access to the ontology (TBox) associated with the data which is usually much smaller than ABox, and relies on SPARQL queries to access the instance data (ABox). ProbAVT can be seen as a kind of topic model [24], [25] or a directed graphical model with hidden variables [21]. In fact, it is also a special case of Infinite Semantic Hidden Models [26] which incorporate expressive ontologies (i.e. $\mathcal{SHOIN}(D)$) to learn predictive models by encoding logic rules as constraints in the variables of Hidden Markov Models. In our case we only encode subclass hierarchies using the constraint $p(x_{kn} \mid a_{kn}) = 0$ if x_{kn} is not a descendent of a_{kn}. This allows the model to be learned from statistical queries and scale to large hierarchies.

C. Discussion

Our approach of using hidden variables to encode subclass hierarchies provides an alternative to the global cut framework proposed by [18], [19] and used in [27]. A cut on a hierarchy is a subset of nodes such that every leaf of the hierarchy is a descendant of some member in the cut. Loosely speaking, a cut defines a *hard* abstraction over a hierarchy, whereas our proposed method defines a *soft* and more general abstraction using hidden variables, recall that we write $p(a_{kn} \mid c, x_{kn})$ to describe the *distribution* of abstractions that best represent a value x_{kn} given class. Another major difference between these two methods is that a cut is seen as a model structure where as hidden variables introduce parameters within the model, and they present different learning challenges. Structure learning for the cut framework requires a search over the cut space, which [19] uses the conditional log likelihood to define the cut refinement criterion; however, introducing hidden variables can make it intractable to estimate the marginal likelihood of parameters (since they are not directly observed from data and their values need to be inferred). Despite these challenges, our experimental results show that the variational Bayesian EM approach [21] can be scaled to large hierarchies over tens of thousands of nodes. Furthermore, with the proposed simple dependency structure (Figure 1), variational Bayesian EM can be executed using only a set of statistical queries from data, which can be obtained from a remote data set through a query interface that supports such queries, which makes the approach useful in settings in which the learning algorithm does not have direct (in memory) access to data.

D. Future Work

In this work we only considered three examples of dependency structures for the new abstraction variables (Fig. 1(b-d)), where we observe differing predictive performances in our experiments. Thus, it would be interesting to investigate model selection in this setting to learn the appropriate model structure from data. It is also interesting to model dependencies between multiple attributes, perhaps using adaptations of a multi-modal topic model [28], [29], [30] in our setting. Specifically, it would be interesting to show that such models or their approximations can be learned using statistical queries against RDF data stores. It would also be interesting to consider extensions of our approach that would allow the use of subproperty hierarchies permitted by RDFS.

VI. Acknowledgments

This work was supported in part by grants (IIS 0711356, CCF 1518732) from the National Science Foundation, the Edward Frymoyer Chair in Information Sciences and Technology held by Vasant Honavar, and the Center for Big Data Analytics and Discovery Informatics co-sponsored by the College of Information Sciences and Technology, the Institute for Cyberscience, the Huck Institutes of the Life Sciences, and the Social Science Research Institute at the Pennsylvania State University.

References

[1] P. Hitzler, M. Krötzsch, and S. Rudolph, *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC, 2009.

[2] R. Cyganiak and A. Jentzsch, "Linking open data cloud diagram." Online. http://lod-cloud.net/, September 2011.

[3] L. Getoor and B. Taskar, *Introduction to Statistical Relational Learning*. The MIT Press, 2007.

[4] Y. Huang, V. Tresp, M. Nickel, A. Rettinger, and H.-P. Kriegel, "A scalable approach for statistical learning in semantic graphs," *Semantic Web*, vol. 5, no. 1, pp. 5–22, 2014.

[5] H. T. Lin and V. Honavar, "Learning classifiers from chains of multiple interlinked RDF data stores," in *Proceedings of the IEEE 2nd International Congress on Big Data*, 2013.

[6] H. T. Lin, N. Koul, and V. Honavar, "Learning relational bayesian classifiers from RDF data," in *Proceedings of the 10th international conference on The semantic web*, vol. 1, pp. 389–404, 2011.

[7] V. Bicer, T. Tran, and A. Gossen, "Relational kernel machines for learning from graph-structured RDF data," in *Proceedings of the 8th Extended Semantic Web Conference*, 2011.

[8] C. Kiefer, A. Bernstein, and A. Locher, "Adding data mining support to SPARQL via statistical relational learning methods," in *Proceedings of the 5th European semantic web conference on The semantic web: research and applications*, p. 478–492, 2008.

[9] J. Neville, D. Jensen, and B. Gallagher, "Simple estimators for relational bayesian classifiers," in *Proceedings of the Third IEEE International Conference on Data Mining*, pp. 609–612, 2003.

[10] D. Brickley and R. Guha, "RDF vocabulary description language 1.0: RDF Schema." Online. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/, Accessed 2004.

[11] W. Cheng, G. Kasneci, T. Graepel, D. H. Stern, and R. Herbrich, "Automated feature generation from structured knowledge," in *CIKM*, pp. 1395–1404, 2011.

[12] S. Bloehdorn and A. Hotho, "Boosting for text classification with semantic features," in *Advances in Web mining and Web usage Analysis*, pp. 149–166, Springer, 2006.

[13] J. Phillips and B. G. Buchanan, "Ontology-guided knowledge discovery in databases," in *K-CAP*, pp. 123–130, 2001.

[14] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006.

[15] G. M. Fung, O. L. Mangasarian, and J. W. Shavlik, "Knowledge-based support vector machine classifiers," in *Advances in neural information processing systems*, pp. 521–528, 2002.

[16] G. M. Fung, O. L. Mangasarian, and J. W. Shavlik, "Knowledge-based nonlinear kernel classifiers," in *Learning Theory and Kernel Machines*, pp. 102–113, Springer, 2003.

[17] O. L. Mangasarian, J. W. Shavlik, and E. W. Wild, "Knowledge-based kernel approximation," *The Journal of Machine Learning Research*, vol. 5, pp. 1127–1141, 2004.

[18] J. Zhang and V. Honavar, "Learning decision tree classifiers from attribute value taxonomies and partially specified data," in *Proceedings of the 2003 Twentieth International Conference on Machine Learning*, ICML '03, (Washington, DC, USA), pp. 880–887, 2003.

[19] J. Zhang, D.-K. Kang, A. Silvescu, and V. Honavar, "Learning accurate and concise naïve bayes classifiers from attribute value taxonomies and data," *Knowl. Inf. Syst.*, vol. 9, pp. 157–179, Feb. 2006.

[20] W3C SPARQL Working Group, "SPARQL 1.1 overview," 2013.

[21] M. J. Beal and Z. Ghahramani, "Variational bayesian learning of directed graphical models with hidden variables," *Bayesian Analysis*, vol. 1, no. 4, pp. 793–832, 2006.

[22] J. Neville, D. Jensen, L. Friedland, and M. Hay, "Learning relational probability trees," in *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 625–630, 2003.

[23] D.-K. Kang, J. Zhang, A. Silvescu, and V. Honavar, "Multinomial event model based abstraction for sequence and text classification," in *Proceedings of the 6th international conference on Abstraction, Reformulation and Approximation*, SARA'05, (Berlin, Heidelberg), pp. 134–148, Springer-Verlag, 2005.

[24] D. Blei and J. Lafferty, "Topic models," *Text mining: classification, clustering, and applications*, vol. 10, p. 71, 2009.

[25] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, pp. 77–84, Apr. 2012.

[26] A. Rettinger, M. Nickles, and V. Tresp, "Statistical relational learning with formal ontologies," in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, ECML PKDD '09, (Berlin, Heidelberg), pp. 286–301, Springer-Verlag, 2009.

[27] N. Bui and V. Honavar, "On the utility of abstraction in labeling actors in social networks," in *Advances in Social Networks Analysis and Mining 2013, ASONAM '13, Niagara, ON, Canada - August 25 - 29, 2013*, pp. 692–698, 2013.

[28] R. Balasubramanyan and W. W. Cohen, "Block-lda: Jointly modeling entity-annotated text and entity-entity links," in *Proceedings of the Eleventh SIAM International Conference on Data Mining*, pp. 450–461, 2011.

[29] J. Chang, J. Boyd-Graber, and D. M. Blei, "Connections between the lines: augmenting social networks with text," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, (New York, NY, USA), pp. 169–178, ACM, 2009.

[30] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen, "Joint latent topic models for text and citations," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, (New York, NY, USA), pp. 542–550, ACM, 2008.