Infinite Hidden Semantic Models for Learning with OWL DL

Achim Rettinger[1] and Matthias Nickles[2]

[1] Technische Universität München, Germany,
achim.rettinger@cs.tum.edu,
[2] University of Bath, UK,
M.L.Nickles@cs.bath.ac.uk

Abstract. We propose a learning model for integrating OWL DL ontologies with statistical learning. In contrast to existing learning methods for the Semantic Web and approaches to the use of prior knowledge in machine learning, we allow for a semantically rich and fully formal representation of rules and constraints which enhance and control the learning task. In our implementation, we achieve this by combining a latent relational graphical model with description logic inference in a modular fashion. To demonstrate the feasibility of our approach we provide experiments with real world data accompanied by a set of $\mathcal{SHOIN}(D)$ axioms. The results illustrate two practical advancements: First, the probability of unknown roles of individuals can be inductively inferred without violating the constraints and second, known ABox axioms can be analyzed by means of clustering individuals per associated concept.

1 Introduction

This paper focuses on the combination of statistical machine learning with OWL DL ontologies and proposes *Infinite Hidden Semantic Models* (IHSM) for this task. The purpose of this is to allow (i) for the completion of the knowledge base with predictions about unknown roles of individuals while considering constraints as background knowledge for the machine learning process and (ii) for the analysis of the known concepts of individuals by means of clustering.

While there is some research on data mining for the Semantic Web (SW), like instance-based learning and classification of individuals, considering constraints specified in the ontology during this tasks has hardly been tried so far. The same applies to the use of "hard" constraints as opposed to the ubiquitous use of "soft" background knowledge in machine learning.

While we use a social network OWL DL ontology as running example, and settle on relational learning as an apparently natural counterpart for logical constraints, our general approach is in no way restricted to these and could be easily adapted to other formal and learning frameworks.

The remainder of this paper is structured as follows: In Section 2 we specify an ontology in OWL DL that defines the taxonomy, relational structure and constraints. Next we show how to infer a relational model from the ontology and transfer the relational model into an IHSM (see Section 3). Then, we learn the parameters of this model in an infinite and unsupervised manner while taking the constraints into account (see Section 4). In Section 5 the IHSM is evaluated empirically using a complex dataset from the semantic web. Finally, we discuss related work and conclude in Section 6.

2 Formal Framework

We settle on the $\mathcal{SHOIN}(D)$ [1] description logic, because ontology entailment in the current Semantic Web quasi-standard OWL DL can be reduced to $\mathcal{SHOIN}(D)$ knowledge base satisfiability. But since we don't make use of any special features our approach could be adapted to any other description language, OWL variant or full first-order logic.

$$\begin{aligned} C \rightarrow A|\neg C|C_1 \sqcap C_2|C_1 \sqcup C_2|\exists R.C|\forall R.C \quad &| \\ \geq nS| \leq nS|\{a_1, ..., a_n\}| \geq nT| \leq nT&| \\ \exists T_1, ..., T_n.D|\forall T_1, ..., T_n.D|D \rightarrow d|\{c_1, ..., c_n\}& \end{aligned}$$

Here, C denote *concepts*, A denote *atomic concepts*, R denote *abstract roles* or *inverse roles* of abstract roles (R^-), S denote abstract *simple roles*, the T_i denote *concrete roles*, d denotes a concrete *domain predicate*, and the a_i / c_i denote abstract / concrete *individuals*.

A $\mathcal{SHOIN}(D)$ *ontology* or *knowledge base* is then a non-empty, finite set of TBox axioms and ABox axioms ("facts") $C_1 \sqsubseteq C_2$ (inclusion of concepts), $Trans(R)$ (transitivity), $R_1 \sqsubseteq R_2$, $T_1 \sqsubseteq T_2$ (role inclusion for abstract respectively concrete roles), $C(a)$ (concept assertion), $R(a, b)$ (role assertion), $a = b$ (equality of individuals), and $a \neq b$ (inequality of individuals). Concept equality is denoted as $C_1 \equiv C_2$ which is just an abbreviation for mutual inclusion, i.e., $C_1 \sqsubseteq C_2, C_2 \sqsubseteq C_1$. Defining a semantics of $\mathcal{SHOIN}(D)$ is not required within the scope of this work, the canonical semantics which we assume in this work can be found, e.g., in [1].

2.1 Constraints

Constraints are actually just knowledge bases (e.g., formal ontologies), and our approach is expected to work with all kinds of logical frameworks which allow for satisfiability (or consistency) checks over some given set of formulas. Formally, we define a set of constraints C to be the deductive closure $\Theta(KB)$ of a given knowledge base KB, with $\Theta(KB) = \{c|KB \models c\}$.
The application-specific constraint set, that we use as an OWL DL ontology is similar to the well-known *Friend-Of-A-Friend* (FOAF) social network schema, together with additional constraints which will be introduced later. The following ontology SN comprises only a fragment of the full FOAF-like ontology we have used (with DOB meaning "date of birth" and $hasBD$ meaning "has birthday".).

$$
\begin{array}{r|c|l}
Person \sqsubseteq Agent & knows^{-} \sqsubseteq knows & \exists knows.\top \sqsubseteq Person \\
\top \sqsubseteq \forall knows.Person & \exists hasBD.\top \sqsubseteq Person & \top \sqsubseteq \forall hasBD.DOB \\
\top \sqsubseteq \leq 1\ hasBD & \top \sqsubseteq \geq 1\ hasBD & \exists yearValue.\top \sqsubseteq DOB \\
\top \sqsubseteq \forall yearValue.gYear & \top \sqsubseteq \leq 1\ yearValue & \exists residence.\top \sqsubseteq Person \\
\top \sqsubseteq \forall residence.Location & \top \sqsubseteq \leq 1\ residence & \exists attends.\top \sqsubseteq Person \\
\top \sqsubseteq \forall attends.School & \exists hasImage.\top \sqsubseteq Person & \top \sqsubseteq \forall hasImage.Image
\end{array}
$$

In addition to these axioms, we provide the machine learning algorithm with an ABox which models an incomplete social network. The later machine learning task consists essentially in a (uncertain) completion of this given network fragment. An example for such additional individuals-governing constraints A:
$tim : Person, tina : Person, tom : Person$
$(tina, tim) : knows, (tina, tom) : knows$

Note that these relationships among persons cannot be weakened or overwritten by the learning process, even if they contradict observed data. They need to be provided manually by the knowledge base engineer. As further constraints, we assume some specific properties G of the analyzed social network. The following set of axioms expresses that every two persons who know each other must share the same chat account provider in case they have a chat account. We present a fragment of the full set:

$$
\begin{array}{r|l}
ProvA \equiv \exists\ HasProv.ProvA & ProvB \equiv \exists\ HasProv.ProvB \\
ProvA \sqsubseteq Person & ProvB \sqsubseteq Person \\
FriendProvA \equiv \exists\ knows.ProvA & FriendProvB \equiv \exists\ knows.ProvB \\
FriendProvA \sqsubseteq Person & FriendProvB \sqsubseteq Person \\
\multicolumn{2}{c}{ProvFrProvA \equiv \exists\ IsProvOf.FriendProvA} \\
\multicolumn{2}{c}{ProvFrProvB \equiv \exists\ IsProvOf.FriendProvB} \\
ProvFrProvA \sqsubseteq ProvA & ProvFrProvB \sqsubseteq ProvB \\
\exists\ knows\ Thing \sqsubseteq Person & \top \sqsubseteq \forall\ knows.Person \\
HasProv \equiv IsProvOf^{-} & \exists\ HasProv\ Thing \sqsubseteq Person \\
\top \sqsubseteq \forall\ HasProv.Prov & HasProv \equiv IsProvOf^{-} \\
\exists\ IsProvOf\ Thing \sqsubseteq Prov & \top \sqsubseteq \forall\ IsProvOf.Person \\
provA : ProvA & provB : ProvB \\
provC : ProvC & provD : ProvD
\end{array}
$$

The complete set of constraints in our running example is then $C = \Theta(SN \sqcup A \sqcup G)$. ($Prov$ stands for "chat account provider".)

Example Data The set of data used as examples for the learning tasks takes the outer form of restricted $\mathcal{SHOIN}(D)$ ABox formulas. But in contrast to the constraints, an example (as a formula) might be wrong, in the sense that it contradicts C. We also do not require the examples to be mutually consistent. In order to maintain compatibility with the expected input format for relational learning, we restrict the syntax of example data to the following two forms:

$$instance : category$$
$$(instance_a, instance_b) : role$$

$\mathcal{SHOIN}(D)$ roles correspond to binary predicates in FOL, and - more or less - naturally to the relations we would like to induce in the following from observed data. The set of all example data as logical formulas is denoted as D.

3 Infinite Hidden Semantic Models

The proposed *Infinite Hidden Semantic Model* (IHSM) is a machine learning algorithm from the area of *Statistical Relational Learning* (SRL) [2]. The novelty is its additional ability to exploit formal ontologies given as a set of logical formulas. In our case, the constraints are provided as a $\mathcal{SHOIN}(D)$ ontology with a T- and an ABox (either can be empty).

In traditional ML prior knowledge is specified implicitly by probability distributions, parameters of the learning algorithm or selection of features. In more advanced models prior believes can be specified by hyperparameters and by the dependency structure in-between random variables. However, we define the relational structure by logical formulas which at the same time impose constraints on the learning.

In this section, we first show how the ontology from Section 2 defines a *Relational Model* (RM). Then we describe how to extend it to an *Infinite Hidden Relational Model* (IHRM) and how to constrain it resulting in the IHSM.

3.1 Relational Models

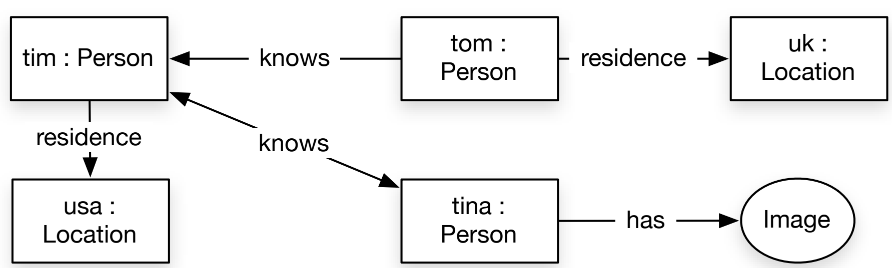

Fig. 1. Partial sociogram of the LJ-FOAF-domain.

In order to predict unknown instances of roles, we need to create an abstract RM of concepts and roles defined in our social network ontology, first. This is done to inform the learning algorithm about which relations to consider. Based on the TBox axioms given by the ontology we can create a simple sociogram as depicted in Fig. 1. A sociogram consists of three different elements: Concept individuals (individuals that are instances of a concept (e.g.

tim : *Person*)), Attribute instances (relations between a concept and a literal (e.g. *Tim* : *hasImage*)), Role instances (relations between concepts (e.g. (*tina*, *tim*) : *knows*))

Concepts, attributes and roles are those defined in the TBox. They build the basis of the RM. Note that many TBox elements first need to be deductively inferred from the ontology, so that all individuals can be assigned to its most specific concepts. This process is known as *Realization* in description logic reasoning. Fig. 2 shows the full RM we use for experiments in Section 5.

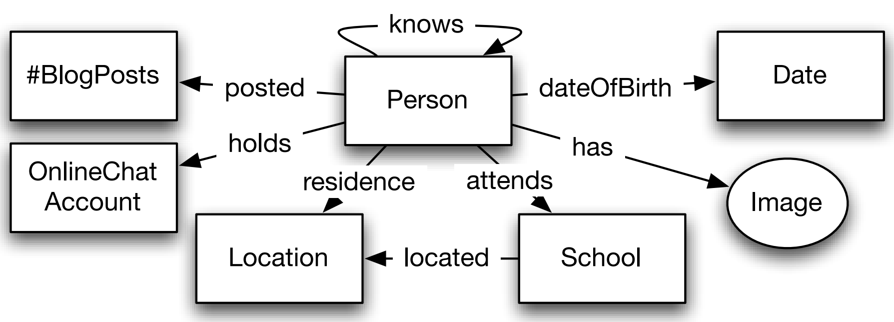

Fig. 2. Relational Model of the LJ-FOAF-domain.

3.2 Infinite Hidden Relational Models

To obtain a probabilistically sound model based on the RM, random variables as well as probability distributions and their parameters need to be introduced next. Following [3] or [4] we extend the RM to a Hidden Relational Model (HRM) by assigning a hidden variable denoted as Z to each concept. The according HRM of the sociogram shown in Fig. 1 is depicted in Fig. 3. Following the idea of hidden variables in *Hidden Markov Models* (HMMs) or *Markov Random Fields* those additional variables can be thought of as unknown properties (roles or attributes) of the attached concept. In addition the hidden variables in the IHSM incorporate restrictions in the form of constraints imposed by the ontology (see Section 3).

Considering the HRM model shown in Fig. 3, information can now propagate via those interconnected hidden variables Z. E.g. if we want to predict whether *tom* Z_3^1 might know *tina* Z_2^1 we need to consider a new relationship $R_{3,2}$. Intuitively, the probability is computed based on (i) the attributes A_3^1 and A_1^1 of the immediately related persons Z_3^1 and Z_2^1; (ii) the known relations associated with the persons of interest, namely the role *knows* and *residence* $R_{2,1}$, $R_{3,1}$ and $R_{3,2}$; (iii) higher-order information and constraints transferred via hidden variables Z_3^1 and Z_2^1. Given that the hidden variables have discrete probability distributions

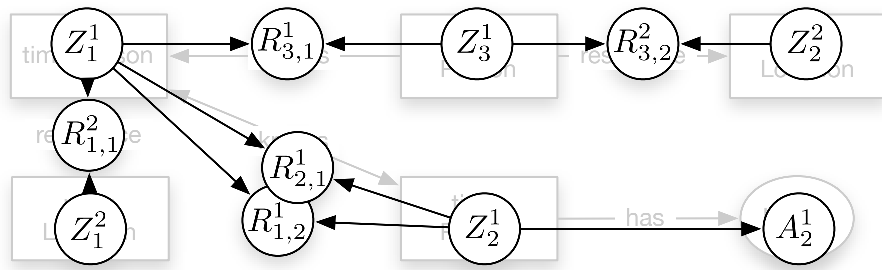

Fig. 3. Hidden relational model of the sociogram defined in Fig. 1.

they can be intuitively interpreted as clusters where similar individuals (in our case similar persons, locations, schools,...) are grouped.

Considering the special importance of the hidden variables in the proposed model their effectiveness highly depends on the number of components that can be represented. Infinite Hidden Relational Models introduced by [3] and[4] offer a solution to this problem. In the IHRM, a hidden variable has a potentially infinite number of states, which have the ability to determine the optimal number of actually occupied components automatically during the inference process.

Finally, we need to define the variables, their probability distributions and parameters. The most important parameters of an IHRM are shown in Fig. 4. The state of Z_i specifies the cluster assignment of the concept i. K denotes the number of clusters in Z. Z is sampled from a multinomial distribution with parameter vector $\pi = (\pi_1, \ldots, \pi_K)$, which specify the probability of a concept belonging to a cluster, i.e. $P(Z_i = k) = \pi_k$. π is referred to as mixing weights, and is drawn according to a stick breaking construction with a hyperparameter α_0. α_0 is referred to as a *concentration parameter* in Dirichlet Process (DP) mixture modeling. It determines the tendency of number of states in Z.

Attributes A^c are generated from a Bernoulli distribution For each component, there is an infinite number of mixture components θ_k. Each person inherits the mixture component, thus we have: $P(G_i = s|Z_i = k, \Theta) = \theta_{k,s}$. These mixture components are independently drawn from a prior G_0. The base distributions G_0^c and G_0^b are conjugated priors with hyperparameters β^c and β^b.

Roles $R_{i,j}$ between two persons (i and j) are a samples drawn from a binomial distribution with parameter $\phi_{k,\ell}$, where k and ℓ denote cluster assignments of the person i and the person j, respectively. $\phi^b_{k,\ell}$ is the correlation mixture component indexed by potentially infinite hidden states k for c_i and ℓ for c_j, where c_i and c_j are indexes of the individuals involved in the relationship class b. Again, G_0^b is the Dirichlet Process base distribution of a role b. If an individual i is assigned to a cluster k, i.e. $Z_i = k$, the person inherits not only θ_k, but also $\phi_{k,\ell}, \ell = \{1, \ldots, K\}$.

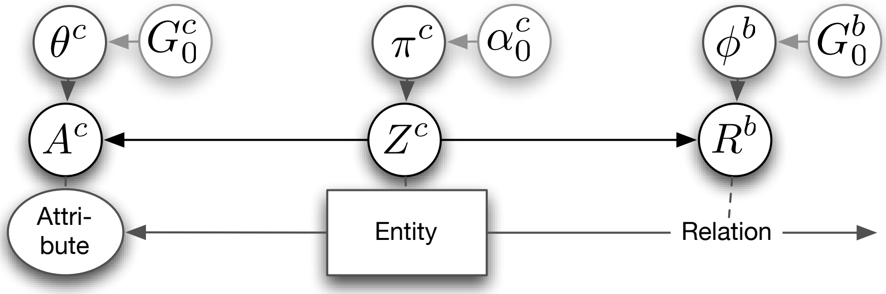

Fig. 4. Parameters of a IHRM.

3.3 Infinite Hidden Semantic Models

The (IHSM) is based on the simple idea that formal constraints can be imposed on the hidden variables. This way a user-designed formal ontology with logical rules can be considered during the learning process. As mentioned before, predictions about attributes and roles depend on the hidden variables Z. If probable predictions for an individual belonging to cluster Z_i are checked against the ontology, it can be determined if individual i is consistent with this cluster according to the constraints. This way valid cluster assignments can be obtained and invalid cluster assignments are excluded.

4 Learning, Constraining and Predictions

The key inferential problem in the IHSM is to compute the joint posterior distribution of unobservable variables given the data. In addition, we need to maintain consistent cluster assignments during learning. As computation of the joint posterior is analytically intractable we use Markov chain Monte Carlo (MCMC) sampling to approximate the posterior distribution. In specific, we apply the blocked Gibbs sampling (GS) with truncated stick breaking representation [5] .

Let D be the set of all available observations (observed example data, each represented as a logical formula as defined in 2.1), and let $Agents = Agent^I$ be the set of all instances of category $Agent$ under interpretation I - that is informally, all persons which contribute to the social network. At each iteration, we first update the hidden variables conditioned on the parameters sampled in the last iteration, and then update the parameters conditioned on the hidden variables. So, for each entity class

1. Update hidden variable Z_i^c for each e_i^c
   (a) Constrain to satisfiable cluster assignments:
   For entity cluster k, let $F_{ext}^k = F^k \cap \{(\mathbf{e_m}, \mathbf{e_n}) : \mathbf{r} | e_m, e_n \in Agents, r \in R, m \neq n\}$ be the set of those logical formulas in the example data set which represent some relation ("role") r between two different individuals

(persons) e_m and e_n where person e_m is assigned to cluster k already and e_n is assigned to a cluster other than cluster k. To keep the notation compact, we spell out role instances $(\mathbf{e_1}, \mathbf{e_2}) : \mathbf{r}$ only asymmetrically (i.e., we omit $(\mathbf{e_2}, \mathbf{e_1}) : \mathbf{r}$ if we have covered the case $(\mathbf{e_1}, \mathbf{e_2}) : \mathbf{r}$). Let $F_k \subseteq D$ be the set of *all* example formulas which have already been used to learn cluster k so far, that is, the subset of the data D which has been used for forming that cluster until now. Let furthermore $\theta(e, D)$ be the set of all sampled formulas in D where the person e appears, i.e., $f \in \theta(e, D)$ iff $f \in D \wedge (f \equiv \mathbf{e} : \mathbf{c} \vee f \equiv (\mathbf{e}, \mathbf{e_x}) : \mathbf{r}$ for some $c \in C$, $e_x \in Agents$ and $r \in R$. We use $\rho(e, j)$ to express that a certain entity e has already been assigned to a certain cluster j. The following steps are now used in order to check whether cluster k is usable w.r.t. the given set of logical constraints C:

i. Identify the largest subset F^k_{clean} of formulas within F^k_{ext} which is consistent with C and the set of example data about person e^c_i:

$$F^k_{clean} \subseteq 2^{F^k_{ext}}, \exists\, \mathcal{I}, \mathcal{I} \models F^k_{clean} \cup \theta(e^c_i, D) \cup C,$$
$$\forall F \subseteq 2^{F^k_{ext}}, \exists\, \mathcal{I}, \mathcal{I} \models F \cup \theta(e^c_i, D) \cup C : F \subseteq F^k_{clean}$$

($\mathcal{I}$ being an interpretation).

ii. Verify whether F^k_{clean}, the formulas which have been used to learn "related" other clusters, $\theta(e^c_i, D)$ and the constraints are consistent in sum <u>if</u> we replace in F^k_{clean} the names of all persons which are assigned to clusters other than k with the name of person e^c_i. Let $F^k_{upd} = \{(\mathbf{e^c_i}, \mathbf{e_m}) : \mathbf{r} | (\mathbf{e_n}, \mathbf{e_m}) : \mathbf{r} \in F^k_{ext}\}$ be the latter set of formulas. Furthermore, let $F^k_{rel} = \bigcup_{j \neq k, \rho(e_m, k), (\mathbf{e_m}, \mathbf{e_n}) : \mathbf{r} \in F^j} F^j$ be the set of all formulas in all other clusters than k which "relate" to cluster k using role formulas. The overall consistency check for cluster k yields a positive result *iff*

$$\exists\, \mathcal{I}, \mathcal{I} \models \theta(e^c_i, D) \cup F^k_{upd} \cup F^k_{rel} \cup C \wedge F^k_{clean} \neq \emptyset$$

(b) Assign to cluster where the consistency check described above yielded a positive result with probability proportional to:

$$\pi^{c(t)}_k P(A^c_i | Z^{c(t+1)}_i = k, \Theta^{c(t)}) \times \prod_{b'} \prod_{j'} P(R^{b'}_{i,j'} | Z^{c(t+1)}_i = k, Z^{c_{j'}(t)}_{j'}, \Phi^{b'(t)})$$

2. Update $\pi^{c(t+1)}$ as follows:

(a) Sample $v^{c(t+1)}_k$ from Beta$(\lambda^{c(t+1)}_{k,1}, \lambda^{c(t+1)}_{k,2})$ for $k = \{1, \ldots, K^c - 1\}$ with

$$\lambda^{c(t+1)}_{k,1} = 1 + \sum_{i=1}^{N^c} \delta_k(Z^{c(t+1)}_i),$$

$$\lambda^{c(t+1)}_{k,2} = \alpha^c_0 + \sum_{k'=k+1}^{K^c} \sum_{i=1}^{N^c} \delta_{k'}(Z^{c(t+1)}_i),$$

and set $v_{K^c}^{c(t+1)} = 1$. $\delta_k(Z_i^{c(t+1)})$ equals to 1 if $Z_i^{c(t+1)} = k$ and 0 otherwise.

(b) Compute $\pi^{c(t+1)}$ as: $\pi_1^{c(t+1)} = v_1^{c(t+1)}$ and

$$\pi_k^{c(t+1)} = v_k^{c(t+1)} \prod_{k'=1}^{k-1} (1 - v_{k'}^{c(t+1)}), \quad k > 1.$$

Before the next iteration we update the parameters:

$$\theta_k^{c(t+1)} \sim P(\cdot|A^c, Z^{c(t+1)}, G_0^c),$$
$$\phi_{k,\ell}^{b(t+1)} \sim P(\cdot|R^b, Z^{(t+1)}, G_0^b).$$

After the GS procedure reaches stationarity the role of interest is approximated by looking at the sampled values. Here, we only mention the simplest case where the predictive distribution of the existence of a relation $R_{i,j}$ between to known individuals i, j is approximated by $\phi_{i',j'}^b$ where i' and j' denote the cluster assignments of the objects i and j, respectively.

5 Experiments

There is almost no work on statistical relational learning with formal ontologies in general and SW data in particular. The lack of experiments on large and complex real world ontologies is not only due to the absence of algorithms but also due to missing suitable datasets. In this section we will present both, a large and complex SW dataset and the methodology of how to apply IHSM in practice. Ultimately, we evaluate the feasibility of our approach by presenting first experimental results with IHSM in this domain.

5.1 Data and Methodology

As mentioned before our core ontology is based on Friend of a Friend (FOAF) data. The FOAF ontology is defined using OWL DL/RDF(S) and formally specified in the FOAF Vocabulary Specification 0.91[3]. Besides that we make use of further concepts and roles that are given in the data. We gathered our FOAF dataset from user profiles of the community website LiveJournal.com[4].

All extracted concepts and roles are shown in Fig. 2. Tab. 1 lists the number of individuals and instantiated roles. As expected for a social networks *knows* is the primary source of information. This real world data set offers both a sufficiently large set of individuals for inductive learning and a formal ontology specified in RDFS and OWL. To demonstrate the full potential of IHSM we additionally add constraints that are not given in the original ontology (see Section 2.1).

To implement all features of IHSM we made use of open source software packages: The semantic web framework Jena[5] is used to load, store and query

[3] http://xmlns.com/foaf/spec/
[4] http://www.livejournal.com/bots/
[5] http://jena.sourceforge.net/

Table 1. No. of individuals, no. of instantiated roles and final number of clusters

Concept	#Indivi.	Roles	#Inst.	#C. IHRM	#C. IHSM
Location	23	*residence*	45	7	9
School	69	*attends*	128	6	6
OnlineChatAccount	4	*holdsAccount*	37	3	4
Person	1258	*knows*	2513	16	19
		hasImage	53		
Date	6	*dateOfBirth*	33	2	2
#BlogPosts	5	*posted*	54	2	2

the ontology and Pellet[6] provides the OWL DL reasoning capabilities. We implemented IHSM and integrated it into this framework. In our experiments the truncation parameter was set to $\#Individuals/4$ and $\alpha_0 = \beta_0 = 10$. We ran 100 iterations each, where 50 iterations are discarded as the burn-in period. As this is intended to provide a feasibility evaluation rather than a performance benchmark we did not carry out any parameter tuning.

5.2 Results

We will now report first results on learning the LJ-FOAF data set. Concerning the computational complexity, the additional consistency check for every individual per iteration is approximately slower by a factor of 6. However, this is partly due to the prototypical implementation of IHSM.

An interesting outcome is the number of clusters per hidden variable after convergence (see Table 1). Considering the difference between IHRM and IHSM, the larger cluster numbers suggest that concepts affected by constraints are likely to be found more diverse by IHSM.

Fig. 5 compares the learned parameter ϕ^{knows} of IHRM to the one learned by IHSM. This indicates the correlation of *Person* clusters with regard to the role *knows*. IHRM converged to 16 und IHSM to 19 clusters. On top of the clusters we show the number of instances per cluster. A brighter cell indicates stronger relations between two clusters. Note that *knows* is considered a directed relation, thus this does not result in a symmetric matrix.

Although hard to generalize, a cell with 50% gray might indicate that no significant probabilistic dependencies for individuals in this cluster are found in the data. Regarding this, it seems to be surprising that *knows*-relations of large clusters predominantly show this characteristics. Interestingly, individuals in those large clusters are those user profiles that are just referenced by other persons but not extracted by the web crawler. Besides that, social networks tend to have a few strongly connected users and many less active users with little connections. This may reinforce this effect.

As mentioned before, IHSM results in more diverse clusters. Most interestingly, there are rows with noticeable darker cells (marked with black arrows)

[6] http://pellet.owldl.com/

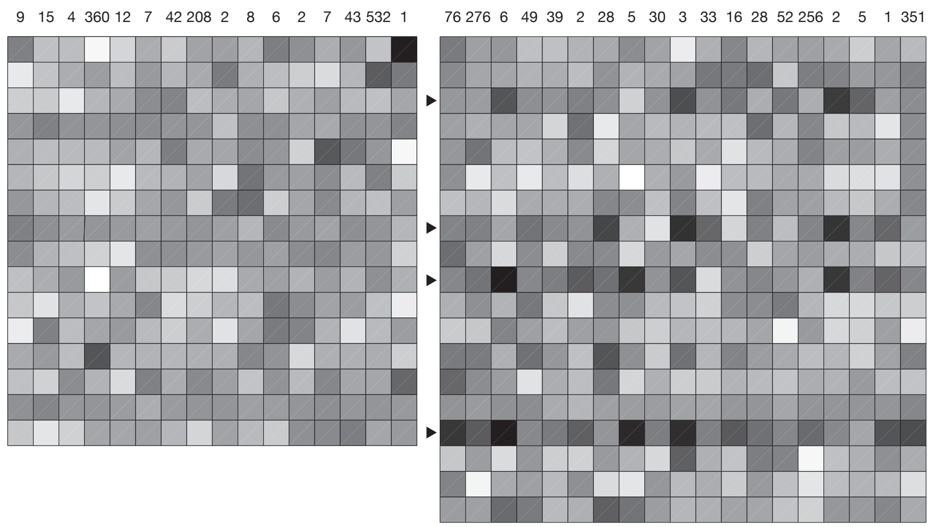

Fig. 5. Correlation mixture component ϕ^{knows} for each combination of clusters Z^{Person}. Left: without constraining (IHRM). Right: with constraining (IHSM). #Individuals per cluster is shown on top.

and those darker cells are symmetric. This suggests that those clusters represent conflicting individuals. In fact, all of those cells contained at least one pair of persons that conflicted with the ontology. This indicates that one of the main goal of IHSM is achieved, namely the exploitation of constraints provided by the ontology.

6 Related Work and Conclusion

Very generally speaking, our proposed method aims at combining machine learning with formal logic. So far, machine learning has been mainly approached either with statistical methods, or with approaches which aim at the inductive learning of formal knowledge from examples which are also provided using formal logic. The most important direction in this respect is *Inductive Logic Programming* (ILP). *Probabilistic*- and *Stochastic Logic Programming* (e.g., [6]) (SLP) are a family of ILP-based approaches which are capable of learning stochastically weighted logical formulas (the weights of formulas, respectively). In contrast to that, our approach learns probability distributions with the help of a given, formal theory which acts as a set of hard constraints. To our best knowledge, this direction is new. Although (S)ILP and statistical relational learning [2] are conceptually very closely related and often subsumed under the general term *relational learning*, statistical relational learning is still rarely integrated with formal logic or ontologies as prior knowledge. One exception are *Markov Logic Networks* (MLN) [7] which combine First Order Logic and *Markov Networks* and learn weights of formulas.

Surprisingly there are also hardly any applications of (pure) SRL algorithms to (SW) ontologies. The few examples, e.g. [8], [9], do not consider formal constraints. The use of *hard constraints* for clustering tasks in purely statistical approaches to learning, as opposed to the ubiquitous use of "soft" prior knowledge, has been approached in, e.g., [10]. A common characteristic of these approaches is that they work with a relatively narrow, semi-formal notion of constraints and do not relate constraints to relational learning. In contrast to these works, our approach allows for rich constraints which take the form of a OWL DL knowledge base (with much higher expressivity). The notion of forbidden pairings of data points (*cannot-link* constraints [10]) is replaced with the more general notion of logical (un-)satisfiability w.r.t. formal background knowledge.

With the presented approach, we hope to open up a new line of future research directions. In general we are curious to see more work on inductive learning with SW ontologies and on the other hand SW ontologies that can be supplemented by uncertain evidence. Concerning IHSM in particular a detailed empirical and theoretical analysis on the effect of constraining on clusters seems promising. We also expect experimental proof for improved predictive performance when formal ontologies are exploited.

References

1. Horrocks, I., Patel-Schneider, P.F.: Reducing owl entailment to description logic satisfiability. In: Journal of Web Semantics, Springer (2003) 17–29
2. Getoor, L., Taskar, B., eds.: Introduction to Statistical Relational Learning. The MIT Press (2007)
3. Kemp, C., Tenenbaum, J.B., Griffiths, T.L., Yamada, T., Ueda, N.: Learning systems of concepts with an infinite relational model. In: Proc. 21st Conference on Artificial Intelligence. (2006)
4. Xu, Z., Tresp, V., Yu, K., Kriegel, H.P.: Infinite hidden relational models. In: Proceedings of the 22nd International Conference on Uncertainity in Artificial Intelligence (UAI 2006). (2006)
5. Ishwaran, H., James, L.: Gibbs sampling methods for stick breaking priors. Journal of the American Statistical Association **96**(453) (2001) 161–173
6. Raedt, L.D., Kersting, K.: Probabilistic logic learning. SIGKDD Explor. Newsl. **5**(1) (2003) 31–48
7. Richardson, M., Domingos, P.: Markov logic networks. Journal of Machine Learning Research **62** (2006) 107–136
8. Kiefer, C., Bernstein, A., Locher, A.: Adding Data Mining Support to SPARQL via Statistical Relational Learning Methods. In: Proceedings of the 5th European Semantic Web Conference (ESWC). Volume 5021 of Lecture Notes in Computer Science., Springer-Verlag Berlin Heidelberg (2008) 478–492
9. N. Fanizzi, C. d'Amato, F.E.: Induction of classifiers through non-parametric methods for approximate classification and retrieval with ontologies. International Journal of Semantic Computing vol.2(3) (2008) 403 – 423
10. Davidson, I., Ravi, S.S.: The complexity of non-hierarchical clustering with instance and cluster level constraints. Data Min. Knowl. Discov. **14**(1) (2007) 25–61