Classification and Retrieval through Semantic Kernels

Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito

Dipartimento di Informatica, Università degli Studi di Bari
{claudia.damato|fanizzi|esposito}@di.uniba.it

Abstract. This work proposes a family of language-independent semantic kernel functions defined for individuals in an ontology. This allows exploiting well-founded kernel methods for several mining applications related to OWL knowledge bases. Namely, our method integrates the novel kernel functions with a *support vector machine* that can be set up to work with these representations. In particular, we present preliminary experiments where statistical classifiers are induced to perform the tasks of instance classification and retrieval.

1 Ontology Mining

Many application domains require operating on large repositories made up of structured data. In the field of the Semantic Web (SW) [2], knowledge intensive manipulations on complex relational descriptions to be performed by machines are foreseen. In this context, expressive languages borrowed from *Description Logics* (DLs) [1] have been adopted for representing ontological knowledge. Unfortunately, machine learning through logic-based methods is inherently intractable in multi-relational settings [13], unless language bias is imposed to constrain the representation, for the sake of tractability, only very simple DLs have been considered [4, 15, 14]. On the other hand, kernel methods [16] represent a family of statistical learning algorithms, including the *support vector machines* (SVMs), that have been effectively applied to a variety of tasks, recently also in domains that typically require structured representations [10, 11]. They can be very efficient since they map, by means of a kernel function, the original feature space into a high-dimensional space, where the learning task is simplified. Such a mapping is not explicitly performed (*kernel trick*): the usage of a positive definite kernel function (i.e. a *valid* kernel) ensures that the embedding into a new space exists and the kernel function corresponds to the inner product in this space [16]. Two components of kernel methods have to be distinguished: the kernel machine and the kernel function. The kernel machine encapsulates the learning task, the kernel function encapsulates the hypothesis language. The same kernel machine can be applied to several knowledge representations, provided a suitable kernel function for each of them.

As argued in [3], most of the research in the SW context address the problem of learning *for* the SW, less attention has been given to the problem of learning *from* the SW data. Looking in this direction, kernel methods can be adopted for several tasks such as classification, clustering and ranking of individuals. In this work, we exploit a kernel method, specifically a SVM, for performing inductive concept retrieval and query answering. To accomplish this goal, a kernel function for DL representation (henceforth

DL-kernel) is proposed. It encodes a notion of similarity of individuals, by exploiting only semantic aspects of the reference representation. Currently, concept retrieval and query answering are performed using merely deductive procedures which easily fail in case of (partially) inconsistent or incomplete knowledge, that can happen when data comes from heterogeneous and distributed resources. We show how the proposed method performs comparably well w.r.t. a standard deductive reasoner, allowing the suggestion of new knowledge that was not previously logically derivable.

In the next section basics of kernel functions for complex representations will be analyzed. In Sect. 3 the *DL-kernel* will be proposed while in Sect. 4 the inductive concept retrieval problem and method will be formally defined. Initial experimental evaluation of the method is presented in in Sect. 5.

2 Kernel Functions

In kernel methods, the learning algorithm (inductive bias) and the choice of the kernel function (language bias) are almost completely independent. The kernel machine encapsulates the learning task, the kernel function encapsulates the hypothesis language. Specifically, the kernel function maps the original feature space of the considered data set into a high-dimensional space, where the learning task is simplified. Such a mapping is not explicitly performed (*kernel trick*): the usage of a positive definite kernel function (i.e. a *valid* kernel) ensures that the embedding into a new space exists, so that the kernel function corresponds to the inner product in this space [16]. In this way, an efficient algorithm for attribute-value instance spaces can be converted into one suitable for structured spaces (e.g. trees, graphs) by merely replacing the kernel function.

Kernels functions are endowed with the closure property w.r.t. many operations, one of them is the convolution [12]: kernels can deal with compounds by decomposing them into their parts, provided that valid kernels have already been defined for them.

$$k_{\mathrm{conv}}(x,y) = \sum_{\substack{\overline{x} \in R^{-1}(x) \\ \overline{y} \in R^{-1}(y)}} \prod_{i=1}^{D} k_i(\overline{x}_i, \overline{y}_i) \tag{1}$$

where R is a composition relationship building a single compound out of D simpler objects, each from a space that is already endowed with a valid kernel. The choice of the function R is a non-trivial task which may depend on the particular application.

On the ground of this property several kernel functions have been defined: for string representations, trees, graphs and other discrete structures [10]. Particularly, in [11] it is shown how to define generic kernels based on type construction, where types are defined in a declarative way. While these kernels are defined as depending on specific structures, a more flexible method is to build kernels parametrized on a uniform representation. Cumby and Roth [5] propose the syntax-driven definition of kernels based on a simple DL representation, the *Feature Description Language*. They show that the feature space blow-up is mitigated by the adoption of efficiently computable kernels. Kernel functions for structured data, parametrized on a description language, allow for the employment of algorithms such as SVM's that can simulate feature generation.

These functions transform the initial representation of the instances into the related active features, thus allowing learning the classifier directly from the structured data.

A notion of kernel for the SW representations has first been proposed in [6]. The kernel function is defined for comparing $\mathcal{ALC}$ concept definitions based on the structural similarity of the AND-OR trees corresponding to the normal form of the input concepts. This kernel is not only structural, since it ultimately relies on the semantic similarity of the primitive concepts on the leaves assessed by comparing their extensions through a set kernel. Moreover, the kernel is actually applied to couples of individuals, after having lifted them to the concept level through realization operators (actually by means of approximations of the most specific concept, see [1]). Since these concepts are constructed on the ground of the same ABox assertions it is very likely that structural and semantic similarity tend to coincide. A more recent definition of kernel functions for individuals in the context of the standard SW representations is reported in [3]. Here the authors define a set of kernels for individuals and for the various types of assertions in the ABox (on concepts, datatype properties, object properties). It is not clear how to integrate such separate building blocks; the preliminary evaluation on specific classification problems regarded single kernels or simple additive combinations.

3 A Family of Kernels for Individuals

In the following we adopt the terminology employed in Description Logics (see [1] for details). We report the basics that are needed for the material in this paper.

Consider a triple $\langle N_C, N_R, N_I \rangle$ made up respectively, of a set of concept names N_C, a set of role names N_R and a set of individual names N_I. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ maps (via $\cdot^{\mathcal{I}}$) such names to the corresponding element subsets, binary relations, and objects of the domain $\Delta^{\mathcal{I}}$. A DL language gives the rules for building more complex concept descriptions based on these building blocks by extending the syntax with specific constructs and providing the related interpretation. The *Open World Assumption* (OWA) is made in the underlying semantics. A *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ contains a *TBox* $\mathcal{T}$ and an *ABox* $\mathcal{A}$. $\mathcal{T}$ is the set of terminological axioms o concept descriptions $C \sqsubseteq D$, meaning $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, where C is the concept name and D is its description. $\mathcal{A}$ contains assertions on the world state, e.g. $C(a)$ and $R(a, b)$, meaning that $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$. Subsumption w.r.t. the models of the KB is the most important inference service. Yet in our case we will exploit *instance checking*, that amounts to deciding whether an individual is an instance of a concept [1].

Grounded on [11], a family of valid kernels for the space X of $\mathcal{ALC}$ descriptions has been proposed [6]. In that case the kernel was defined for pairs of concepts descriptions, then the individuals had to be lifted to the concept level before computation (exploiting approximations of the respective most specific concepts w.r.t. the ABox). The main limitation of the kernel is represented by the dependency on the DL language. In order to overcome this limitation, we propose a set of kernels that can be applied directly to individuals, based on ideas exploited for a family of inductive distance measures [9]. The rationale for this kernels is that similarity between individuals is decomposed along with the similarity w.r.t. each concept in a given committee of features (class definitions). Two individuals are maximally similar w.r.t. a given concept F_i if they exhibit

the same behavior, i.e. both are instances of the concept or of its negation. Conversely, the minimal similarity holds when they belong to opposite concepts. Because of the OWA, sometimes a reasoner cannot assess the concept-membership, hence we assign an intermediate value to reflect such uncertainty. The kernel function is formally defined in the following:

Definition 3.1 (family of kernels). *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a KB. Given a set of concept descriptions $\mathsf{F} = \{F_1, F_2, \ldots, F_m\}$, a family of kernel functions $k_p^{\mathsf{F}} : \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mapsto [0, 1]$ is defined as follows:*

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

where $p > 0$ and $\forall i \in \{1, \ldots, m\}$ the similarity function *σ_i is defined: $\forall a, b \in \mathsf{Ind}(\mathcal{A})$*

$$\sigma_i(a, b) = \begin{cases} 1 & (F_i(a) \in \mathcal{A} \wedge F_i(b) \in \mathcal{A}) \vee (\neg F_i(a) \in \mathcal{A} \wedge \neg F_i(b) \in \mathcal{A}) \\ 0 & (F_i(a) \in \mathcal{A} \wedge \neg F_i(b) \in \mathcal{A}) \vee (\neg F_i(a) \in \mathcal{A} \wedge F_i(b) \in \mathcal{A}) \\ \frac{1}{2} & otherwise \end{cases}$$

Note that the kernel functions can be model-theoretically defined by substituting,in the formulation above, the following expression $\mathcal{K} \models F_i(a)$ to $F_i(a) \in \mathcal{A}$ (resp. $\mathcal{K} \models \neg F_i(a)$) to each belongingness expression $F_i(a) \in \mathcal{A}$ (resp. $\neg F_i(a) \in \mathcal{A}$), both for a and b.

Instance-checking is to be employed for assessing the value of the simple similarity functions. This is known to be computationally expensive (also depending on the specific DL language of choice). Alternatively, especially for largely populated ontologies, a simple look-up may be sufficient, as suggested by the formal definition of the σ_i functions. If it is required that $k(a, b) = 0$ even though the selected features are not able to distinguish the individuals, one might make the *unique names assumption* on all individuals occurring in the ABox $\mathcal{A}$, and employ a special feature based on equality: $\sigma_0(a, b) = 1$ iff $a = b$ (and 0 otherwise). Alternatively, equivalence classes might be considered instead of individuals.

The most important property of a kernel function is its validity (it must correspond to a dot product in a certain embedding space).

Proposition 3.1 (validity). *Given an integer $p > 0$ and a committee of features F, the function k_p^{F} is a valid kernel.*

This result can be assessed by proving the function k_p^{F} definite-positive. Alternatively it is easier to prove the property by showing that the function can be obtained by composing simpler valid kernels through operations that guarantee the closure w.r.t. this property [12]. Specifically, since the various σ_i functions ($i = 1, \ldots, n$) correspond to a *matching kernel*, the property follows from the closure w.r.t. sum, multiplication by a constant and kernel multiplication.

It is also worthwhile to note that this is indeed a family of kernels parametrized on the choice of features. As for the semantic pseudo-metric that inspired the kernel definition [9], a preliminary phase may concern finding an optimal choice of features. This may be carried out by means of randomized optimization procedures, similar to the developed for the pseudo-distance [8].

4 Inductive Classification and Retrieval through Kernel Methods

SVMs are classifiers based on kernel functions that, exploiting a kernel function, map the training data into a higher dimensional feature space where they can be classified by means of a linear classifier. This is done by constructing a separating hyperplane with the maximum margin in the new feature space, which yields a nonlinear decision boundary in the input space. By the use of a kernel function, the separating hyperplane is computed without explicitly carrying out the mapping into the feature space. A SVM, as any other kernel method, can be applied to whatever knowledge representation, provided a valid kernel function suitable for the chosen representation. Given the *DL-kernel* defined in Sect. 3, we use the SVM to solve the following problem:

Definition 4.1 (classification problem). *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a KB, let $\mathsf{Ind}(\mathcal{A})$ be the set of all individuals in $\mathcal{A}$ and let $C = \{C_1, \ldots, C_s\}$ be the set of all concepts (both primitive and defined) in $\mathcal{T}$. The* classification problem *to solve is:* ***given*** *an individual $a \in \mathsf{Ind}(\mathcal{A})$,* ***determine*** *the set of concepts $\{C_1, \ldots, C_t\} \subseteq C$ to which a belongs to.*

In the general setting of SVMs, the classes, w.r.t. which the classification is performed, are disjoint. This is not generally verified in the SW context, where an individual can be instance of more than one concept. To solve this problem, a different answering procedure is proposed. The multi-class classification problem is decomposed into smaller binary classification problems (one per class). Specifically, given an individual in the KB, instead of returns the set of concepts to which it belongs to, it is classified w.r.t. to each concept, namely for each concept, the classifier assesses if the individual is instance or not. Therefore, a simple binary value set ($V = \{-1, +1\}$) can be employed, where $(+1)$ indicates that an individual x_i is instance of the concept C_j; (-1) indicates that x_i is not instance of C_j. Anyway, this is not enough. Indeed, in the general setting, an implicit *Closed World assumption* (CWA) is made, while in the SW context, the *Open World Assumption* (OWA) is usually adopted. To deal with the OWA, the absence of information on whether a certain individual x_i belongs to the extension of the concept C_j should not be interpreted negatively, as seen before, rather, it should count as neutral information. Thus, we consider another value set: $V = \{+1, -1, 0\}$, where the three values denote, respectively, assertion occurrence ($C_j(x_i) \in \mathcal{A}$), occurrence of the opposite assertion ($\neg C_j(x) \in \mathcal{A}$) and assertion absence in $\mathcal{A}$. Hence, given a query instance x_q, for every concept $C_j \in C$, the classifier will return $+1$ if x_q is an instance of C_j, -1 if x_q is an instance of $\neg C_j$, and 0 otherwise. The classification is performed on the ground of a set of training examples from which such information can be derived.

Considered a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, all the individuals in $\mathcal{A}$ can be classified w.r.t. one or more concepts in $\mathcal{T}$, thus performing the concept retrieval inductively. In the same way, the classifier can be employed for solving a query answering task, by determining the extension of a new concept built from concepts and roles in $\mathcal{T}$. As it will be shown in the next section, the classifier behavior, in performing concept retrieval, is comparable with the one of a standard reasoner. Moreover, the classifier may be able to induce new knowledge that is not logically derivable.

Table 1. Ontologies employed in the experiments.

ontology	*DL*	*#concepts*	*#obj. prop*	*#data prop*	*#individuals*
S.-W.-M.	$\mathcal{ALCOF}(D)$	19	9	1	115
SCIENCE	$\mathcal{ALCIF}(D)$	74	70	40	331
NTN	$\mathcal{SHIF}(D)$	47	27	8	676
WINES	$\mathcal{ALCIO}(D)$	112	9	10	149

5 Experimental Evaluation

For performing the classification problem defined in Sect. 4 and for assessing the validity of the *DL-kernel* (see Def. 3.1), a SVM from the LIBSVM library[1] has been considered. The instance classification has been performed on four different OWL ontologies: SURFACE-WATER-MODEL, NEWTESTAMENTNAMES, SCIENCE, and WINES from the Protégé library[2]. Details about such ontologies are reported in Tab. 1. The classification method was applied to all the individuals in each ontology; namely, for each ontology, the individuals were checked to assess if they were instances of the concepts in the ontology through the SVM and the *DL-kernel*[3] embedded in it. A similar experimental setting has been considered in [3] with an exemplified version of the GALEN Upper Ontology[4]. The ontology has been randomly populated and only seven concepts have been considered while no roles have been taken into account[5]. Differently from this case, we did not apply any changes on the considered ontologies.

The SVM-based classifier performance was evaluated by comparing its responses to those returned by a standard reasoner[6] used as baseline. The experiment has been performed by adopting the ten-fold cross validation procedure. For each concept in the ontology, the following parameters have been measured for the evaluation: *match rate*: number of cases of individuals that got exactly the same classification by both classifiers with respect to the overall number of individuals; *omission error rate*: amount of unlabeled individuals (namely the method returns 0 as classification results) while they were to be classified as instances of the considered concept; *commission error rate*: amount of individuals labeled as instances of a concept, while they (logically) belong to the negation of that concept or vice-versa; *induction rate*: amount of individuals that were found to belong to a concept or its negation, while this information is not logically derivable from the knowledge base. The average rates obtained over all the concepts in each ontology are reported in Tab. 2, jointly with their range.

Looking at the table, it is important to note that, for every ontology, the commission error is null. This means that the classifier did not make critical mistakes, i.e. cases

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

[2] See the webpage: http://protege.stanford.edu/plugins/owl/owl-library

[3] The feature set for computing the *DL-kernel* was made by all concepts in the considered ontology (see Def. 3.1).

[4] http://www.cs.man.ac.uk/ rector/ontologies/simple-top-bio/

[5] Due to the lack of information for replicating the ontology used in [3], a comparative experiment with the proposed kernel framework cannot be performed.

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

Table 2. Results (average and range) of the experiments using *DL-kernel*.

ONTOLOGY		*match rate*	*induction rate*	*omis. err. rate*	*comm. err. rate*
WINES	*avg.*	**0.952**	**0.006**	**0.042**	**0.000**
	range	0.585 - 0.993	0.000 - 0.415	0.000 - 0.343	0.000 - 0.00
SCIENCE	*avg.*	**0.971**	**0.018**	**0.011**	**0.000**
	range	0.947 - 1.000	0.000 - 0.053	0.000 - 0.021	0.000 - 0.000
S.-W.-M.	*avg.*	**0.959**	**0.000**	**0.041**	**0.000**
	range	0.836 - 1.000	0.000 - 0.000	0.000 - 0.164	0.000 - 0.000
N.T.N.	*avg.*	**0.982**	**0.002**	**0.016**	**0.000**
	range	1.000 - 0.932	0.000 - 0.055	0.000 - 0.068	0.000 - 0.000

Table 3. Results (average and range) of the experiments with $\mathcal{ALC}$ kernel ($\lambda = 1$).

ONTOLOGY		*match rate*	*induction rate*	*omis. err. rate*	*comm. err. rate*
WINES	*avg.*	**0.956**	**0.004**	**0.040**	**0.000**
	range	0.650 - 1.000	0.000 - 0.270	0.010 - 0.340	0.000 - 0.000
SCIENCE	*avg.*	**0.942**	**0.007**	**0.051**	**0.000**
	range	0.800 - 1.00	0.000 - 0.040	0.000 - 0.200	0.000 - 0.000
S.-W.-M.	*avg.*	**0.871**	**0.067**	**0.062**	**0.000**
	range	0.570 - 0.980	0.000 - 0.420	0.000 - 0.400	0.000 - 0.000
N.T.N.	*avg.*	**0.925**	**0.026**	**0.048**	**0.001**
	range	0.660 - 0.990	0.000 - 0.320	0.000 - 0.220	0.000 - 0.030

when an individual is deemed as an instance of a concept while it really is an instance of the negation of that concept. Furthermore, a very high match rate is registered for every ontology. Particularly, looking at Tab. 2, it can be observed that the match rate increases with the increase of the number of individuals in the considered ontology. This is because, being the SVM a statistical method, its performance improves with the augmentation of the set of the available examples. Almost always the SVM-based classifier is able to induce new knowledge. However, a conservative behavior has been also registered, indeed the omission error rate is not null (even if it is very close to 0). To decrease the tendency to a conservative behavior of the method, a threshold could be introduced for the consideration of the "unknown" (labeled with 0) training examples.

We have compared the results obtained by performing the inductive concept retrieval exploiting the *DL-kernel*, with the one obtained applying the SVM-based classifier jointly with the $\mathcal{ALC}$ kernel (see [6, 7]). The outcomes of the second set of experiments are reported in Tab. 3. By comparing Tab. 2 and Tab. 3 it is possible to note that *DL-kernel* improves both match rate and omission rate with respect to $\mathcal{ALC}$ kernel. Consequently a decrease of the induction rate is observed. The commission rate for the $\mathcal{ALC}$ kernel is almost null as for the *DL-kernel*.

6 Conclusions

We have defined a novel family of semantic kernel functions for individuals in the context of populated ontologies based on their behavior w.r.t. a number of features (concept definitions). The kernels are language-independent (they simply require instance-checking or ABox look-up) and can be easily integrated with a kernel machine (a SVM in our case) for performing a broad spectrum of activities. In this paper we focused on the application to statistical classification of individuals in an ontology. The resulting classifier has been used to perform instance classification in an inductive way which

may be more efficient and noise-tolerant w.r.t. the standard deductive procedures. It has been experimentally shown that its performance is not only comparable to the one of a standard reasoner, but the classifier is also able to induce new knowledge, which is not logically derivable. Particularly, an increase in prediction accuracy was observed when the instances are homogeneously spread, as expected from statistical methods. The realized classifier can be exploited for predicting/suggesting missing information about individuals, thus completing large ontologies.

References

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

[2] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[3] S. Bloehdorn and Y. Sure. Kernel methods for mining instance data in ontologies. In Karl Aberer et al., editor, *Proc. of the 6th Int. Semantic Web Conf.*, volume 4825 of *LNCS*, pages 58–71. Springer, 2007.

[4] W.W. Cohen and H. Hirsh. Learning the CLASSIC description logic. In P. Torasso et al., editor, *Proc. of the 4th Int. Conf. on the Principles of Knowledge Representation and Reasoning*, pages 121–133. Morgan Kaufmann, 1994.

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

[6] N. Fanizzi and C. d'Amato. A declarative kernel for $\mathcal{ALC}$ concept descriptions. In F. Esposito et al., editor, *Proc. of the 16th Int. Symposium on Methodologies for Intelligent Systems.*, volume 4203 of *LNCS*, pages 322–331. Springer, 2006.

[7] N: Fanizzi and C. d'Amato. Inductive concept retrieval and query answering with semantic knowledge bases through kernel methods. In F. Esposito et al., editor, *Proc. of Knowledge-Based Intelligent Information and Engineering Systems.*, volume 4692 of *LNCS*, pages 148–155. Springer, 2007.

[8] N. Fanizzi, C. d'Amato, and F. Esposito. Evolutionary conceptual clustering of semantically annotated resources. In *Proc. of the Int. Conf. on Semantic Computing*. IEEE, 2007.

[9] N. Fanizzi, C. d'Amato, and F. Esposito. Induction of optimal semi-distances for individuals based on feature sets. In D. Calvanese et al., editor, *Working Notes of the 20th Int. Description Logics Workshop*, volume 250 of *CEUR Workshop Proceedings*, 2007.

[10] T. Gärtner. A survey of kernels for structured data. *SIGKDD Explorations*, 5(1):49–58, 2003.

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

[12] D. Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, Department of Computer Science, University of California – Santa Cruz, 1999.

[13] D. Haussler, M. J. Kearns, and R. E. Schapire. Bounds on the sample complexity of bayesian learning using information theory and the vc dimension. *Machine Learning*, 14(1):83–113, 1994.

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

[15] J. Lehmann. Concept learning in description logics. Master's thesis, Dresden University of Technology, 2006.

[16] B. Schölkopf and A.J. Smola. *Learning with Kernels*. The MIT Press, 2002.