Description Logics and Knowledge Discovery of Data

Stefan Schlobach*

Introduction. Description Logics (DL) are knowledge representation formalisms which have been applied in a number of application areas over the last decade. Knowledge Discovery on the other hand is becoming more and more popular given the enormous amount of available data and has considerably increased the interest in machine learning techniques for data mining purposes. We have tried to make this machinery available to DL reasoning systems, which enables easy integration of predefined knowledge and data for the Knowledge Discovery process. The core of this new approach, i.e. the formal definition of a learning paradigm in Description Logics and a general framework to calculate *learned concepts* was presented in [Sch00].

Knowledge Discovery from Data. Fayyad et al. define Knowledge Discovery in Databases (KDD) as *the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [FPSS96].* The KDD process consists of three steps. First, data is preprocessed, i.e., interesting and typical (and hopefully crisp and un-noisy) data is chosen. The second step is the data mining process. Data Mining means the application of data analysis and learning algorithms to produce enumerations of patterns over the data. Finally the learned results have to be evaluated and turned into a useful (e.g. human-readable) form.

Description Logics. Description Logics (DL) are set description languages that have been in use for knowledge representation for more than a decade. The hybrid character of reasoning about concepts and about individuals is nicely matched in the hybrid architecture of DL-systems which usually incorporate separate mechanisms for reasoning with extensional and intensional knowledge. The general ideas of A–Box mining are independent of a particular description logic. For this reason we define the set of languages $\mathcal{DL}$ to which our approach is applicable as the set of description logics containing at least conjunction, atomic negation and universal quantification over roles where subsumption, consistency and instance checking is decidable.

Learning and Mining in Description Logics. The main objective of A–Box mining is to generate concepts from a given A–Box $\mathcal{A}$ that can replace the decisions in the D–Box for classification. We will introduce and define two necessary criteria (called covering and exclusiveness) which define our notion of concepts *learned* from $\mathcal{A}$ and $\mathcal{D}$ (and possibly $\mathcal{T}$). These concepts will be called *generalised decision concepts* (GDCs).

The first condition states that all objects in the A–Box that are instances of a GDC learned from a decision D must be instances of D, i.e., all A–Box instances a of a GDC $\underline{\mathcal{A}}D_i$ must be covered by the corresponding decision D_i: i.e. $a \in_{\mathcal{A}} \underline{\mathcal{A}}D_i \Rightarrow a \in_{\mathcal{A}} D_i$. This condition ensures correctness of the classification through the GDC with respect to the original decision. The second condition denotes the fact that any A–Box object a, which is an instance of $\underline{\mathcal{A}}D_i$, must by unambiguously decidable with the corresponding original decision D_i only: $a \in_{\mathcal{A}} \underline{\mathcal{A}}D_i \Rightarrow a \notin_{\mathcal{A}} D_j$. A *generalised decision concept* (GDC) for a decision D is a concept, which is exclusively $\mathcal{A}$-covered by D.

Simplicity and Optimality. The definition for GDCs gives necessary but not sufficient criteria to justify the quality of learned concepts. Depending on the chosen language, there might be infinitely many GDCs for a decision. There are two further criteria which we introduce informally, namely that GDCs should be "as general as possible" (semantic optimality) and "as simple as possible" (syntactic simplicity).

*Dept. of Computer Science, King's College London, Strand, London WC2R 2LS, UK, schlobac@dcs.kcl.ac.uk

Description Logic based KDD Systems. We present a framework for Knowledge Discovery for Description Logics $\mathcal{DL}$ which was investigated in [Sch00] in more detail.

Input:	An A–Box $\mathcal{A}$, T–Box $\mathcal{T}$ and a set of decisions $\{D_1, \ldots, D_n\}$.
Output:	The Set of all evaluated generalised decision concepts for each decision.

- **Preprocessing:** There are three preprocessing steps to be performed that are useful independent of a particular learning or mining method. *A–Box classification* transforms the assertional information about data into related logical concepts. By creating hierarchies for the logical descriptions, *T–Box classification* allows to deal with sparse A–Boxes and missing data. *Separation* of concepts into approximations is a natural mechanism to deal with rough and noisy information.
- **Generalisation:** There are different ways to learn concepts from examples and counterexamples, but mostly a combination of concept expansion and inconsistency check is involved. *Expansion* is the process of choosing superconcepts for relevant logical descriptions, which are inconsistent with logical descriptions (or expansions) of the counterexamples. This part of the framework is in general non constructive.
- **Optimality:** For each particular description logic optimality has to be defined separately. For some languages (e.g. $\mathcal{AL}^{\mathbb{R},=}$) the notions of optimality and simplicity coincide, and sound and complete mechanisms to calculate optimal GDCs can be defined. For other languages and learning methods, the quality of notions for "general and simple" will be more difficult to assess and heuristics have to be incorporated into the learning process.
- **Evaluation of Results:** The evaluation of the results is an integral parts of the KDD process If generalised decision concepts are to become part of the intensional knowledge of the knowledge representation system, they have to be assessed using quantitative accuracy measures and by human experts. Classically, learning results are difficult to read and understand, so that it must be one of the features of a knowledge discovery system to transform the results into a human-readable form. One possible approach is described in [BKM] where Baader et al. investigate algorithms for rewriting concepts using terminologies for several logics in $\mathcal{DL}$.

Implementation. The Group of Logic and Computation at King's College London is currently implementing a family of hybrid systems[1] which enables data driven logic reasoning. Implementations of specialised and efficient algorithms for $\mathcal{AL}^{\mathbb{R}}$, $\mathcal{ALC}$ with concrete domains and $DL_{arc\forall}$ [Ohl99], which allows for an expressive T–Box with a strong arithmetical component and a database–like assertional component are currently under way. The integration of a knowledge discovery facility will be an integral part of the system.

References

[BKM] F. Baader, R. Küster, and R. Molitor. Rewriting concepts using terminologies. Technical Report LTCS-Report 99-12, LuFG Theoretical Computer Science, RWTH Aachen, Germany.

[FPSS96] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. Knowledge discovery and data mining toward a unifying framework. In *Proceeding of The Second Int. Conference on Knowledge Discovery and Data Mining*, pages 82–88, 1996.

[Ohl99] H.J. Ohlbach. A theory-resolution style A–Box calculus. In *M4M Methods for Modalities, Workshop Proceedings*. University of Amsterdam, 1999.

[Sch00] S. Schlobach. Assertional mining in description logics. 2000 International Workshop on Description Logics - DL2000, August 2000.

[1] More information about the `Wellington` knowledge representation and reasoning system and its implementation is available from the author or on: `http://www.dcs.kcl.ac.uk/research/groups/logic/wellington`.