A Boxology of Design Patterns for Hybrid Learning and Reasoning Systems

Frank van Harmelen, Annette ten Teije

Vrije Universiteit Amsterdam
{frank.van.harmelen,annette.ten.teije}@vu.nl

Abstract. We propose a set of design patterns to describe a large variety of systems that combine statistical techniques from machine learning with symbolic techniques from knowledge representation. As in other areas of computer science (knowledge engineering, software engineering, process mining), such design patterns help to systematize the literature, clarify which combinations of techniques serve which purposes, and encourage re-use of software components. We have validated our compositional design patterns against a large body of recent literature.[1]

Recent years have seen a strong increase in interest in combining Machine Learning methods with Knowledge Representation methods, fuelled by the complementary functionalities of both types of methods, and by their complementary strengths and weaknesses. This increasing interest has resulted in a large volume of diverse papers in a variety of venues, and from a variety of communities. This paper is an attempt to create structure in this large, diverse and rapidly growing literature. We present a conceptual framework, in the form of a set of design patterns, that can be used to categorize techniques for combining learning and reasoning. In the full paper, we have validated our design patterns against more than 50 papers from the research literature from the last decade. Our claim is that each of the systems that we encountered in those references is captured by one of our design patterns. Broadly recognized advantages of such design patterns are: they distill previous experience in a reusable form for future design activities, they encourage re-use of code, they allow composition of such patterns into more complex systems, they provide a common language in a community, and they are a useful didactic device

Our design patterns are expressed in a graphical notation[2]. We use ovals to denote algorithmic components that perform some computation, and boxes to denote their input and output. We distinguish two types of algorithmic components (ovals): those that perform some form of deductive inference (labelled as

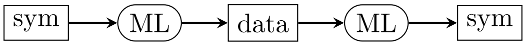
[1] **Full version in J. of Web Engineering, (18),1-3,97-124, 2019.**

[2] *boxology*: "A representation of an organized structure as a graph of labelled nodes and connections between them", https://www.definitions.net/definition/boxology

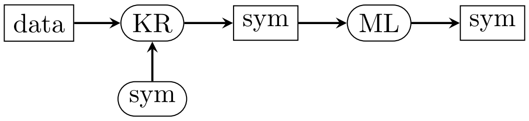
the "KR" components) and those that perform some form inductive inference (the "ML" components): KR ML . We also use two kinds of input- and output-boxes: symbolic structures, and other data: sym data .

We now present some example patterns for hybrid systems that perform reasoning and learning. The full paper presents a larger set of 15 patterns.

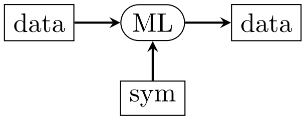
From symbols to data and back again A recent class of "graph completion" systems [3] apply inductive techniques to a knowledge graph to predict addition edges. Almost all graph completion algorithms first translate the knowledge graph to a representation in a high-dimensional vector space (a process called "embedding"), described by the following pattern:



Deriving an intermediate abstraction for reasoning In [2] a raw datastream is first abstracted into a stream of symbols with the help of a symbolic ontology, and this stream of symbols is then fed into a classifier (which performs better on the symbolic data than on the original raw data).



Learning with symbolic information as a prior The following design pattern describes machine learning systems that use prior knowledge:



An example of this are the Logic Tensor Networks in [1], where the authors show that encoding prior knowledge in symbolic form allows for better learning results on fewer training data, as well as more robustness against noise.

Concluding comments Each design pattern abstracts from specific mathematical and algorithmic details of the specific components, and only looks at the functional behaviour of the pattern and the functional dependencies between the components. This makes our descriptions of hybrid systems as design patterns abstract and general.

References

1. Donadello, I., Serafini, L., d'Avila Garcez, A.S.: Logic tensor networks for semantic image interpretation. In: IJCAI. pp. 1596–1602 (2017)
2. Kop, R., et al.: Predictive modeling of colorectal cancer using a dedicated preprocessing pipeline on routine electronic medical records. Comp. in Bio. and Med. **76**, 30–38 (2016)
3. Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic Web **8**(3), 489–508 (2017)