Using Ontological Reasoning and Planning for Data Mining Workflow Composition

Monika Žáková[1], Petr Křemen[1], Filip Železný[1], Nada Lavrač[2]

[1] Czech Technical University in Prague, Czech Republic
[2] Jožef Stefan Institute, Ljubljana, Slovenia, and the University of Nova Gorica, Nova Gorica, Slovenia

Abstract. This paper addresses the problem of semi-automatic design of workflows for complex knowledge discovery tasks. Assembly of optimized knowledge discovery workflows requires awareness of and extensive knowledge about the principles and mutual relations between diverse data processing and mining algorithms. We aim at alleviating this burden by automatically proposing workflows for the given type of inputs and required outputs of the discovery process. The methodology adopted in this study is to define a formal conceptualization of knowledge types and data mining algorithms and design a planning algorithm, which extracts constraints from this conceptualization for the given user's input-output requirements. We demonstrate our approach in two use cases, one from scientific discovery in genomics and another from advanced engineering.

1 Introduction

Integration of heterogeneous data sources and inferring new knowledge from such combined information is one of the key challenges in present-day life science. Consider e.g. bioinformatics where for virtually any biological entity (a gene, for example), vast amounts of relevant background information are available from public web resources. This information comes in diverse formats and at diverse levels of abstraction. Continuing the genomic example, the publicly available data sources range from DNA sequence information, homology and interaction relations, gene-ontology annotations, information on the involvement in biological pathways, expression profiles in various situations etc. To merge only these exemplary sources of data, one already has to combine specialized algorithms for processing sequences, relational data, ontology information and graph data. It is thus no surprise that a principled fusion of such relevant data requires the interplay of diverse specialized algorithms resulting in highly intricate workflows. While the mutual relations of such algorithms and principles of their applicability may be mastered by computer scientists, their command cannot be expected from the end user, e.g. a life scientist.

The primary hypothesis investigated in our study is that such complex scientific workflows can be assembled automatically with the use of a knowledge discovery ontology and a planning algorithm accepting task descriptions automatically formed using the vocabulary of the ontology.

2 Related Work

Several previous works have explored planning in the context of workflows. Notably, within the Pegasus project [5] a planner is used to construct a concrete workflow out of an abstract workflow. In our research we tackle a related yet different goal; given an ontology and a task description, we use a planner to construct a workflow, which in the terminology of [5] would be called abstract. The paper [8] is relevant to our work as it elaborates a procedure for converting OWL-S service annotations into action descriptions in the standard Planning Domain Definition Language (PDDL). Constructing a PDDL problem description is also a technical ingredient of our methodology.

Unlike in [8] we conduct workflow composition tasks in the specific domain on data mining, for which we devise a special ontology. A similar aim was followed by recent work of Brezany et al. [4]. This work, however, is focused only on automatic formation of linear sequences of tasks: their ontology ensures that there is only one algorithm that can be inserted into a workflow prior to another algorithm. In our work we try to provide a more principled framework for the domain of data mining, aimed at enabling the construction of much more complex workflows with the main intended application in non-trivial scientific discovery tasks.

To the best of our knowledge, there is so far no previous work providing a principled and actionable ontology for data mining including relational data mining with complex background knowledge. There have been efforts to provide a systematic description of data and processes for the classical data mining tasks e.g. in systems MiningMart [10], CAMLET [13], CITRUS [17] and NExT [2].

The MiningMart system [10] focuses on propositional data mining from data stored in a relational database. It contains a meta-model for representing and structuring of information about data and algorithms, however this meta-model is expressed in XML, not in an ontology language. The system also does not provide means for automatic workflow creation.

The project CITRUS [17] uses an object oriented schema is used to model relationships between the algorithms. The system focuses on guiding the user through mostly manual process of building of workflows by including information about properties and usability of the algorithm in the algorithm description. Planning is used only for proposing steps in process decomposition and refinement.

In the CAMLET system an ontology of algorithms (processes) and ontology of data structures are defined, however no ontology language is specified in [13]. The system relies on manually defined top-level control structure, which is then refined using genetic programming until a suitable structure producing the required results is found. The structure of algorithms ontology does not attempt to formalize the domain systematically, rather it is determined by the used top-level control structure.

The most systematic effort to construct a general knowledge discovery ontology is described in [2]. The ontology used by the NExT system is built on OWL-S and provides a relatively detailed structure of the propositional data

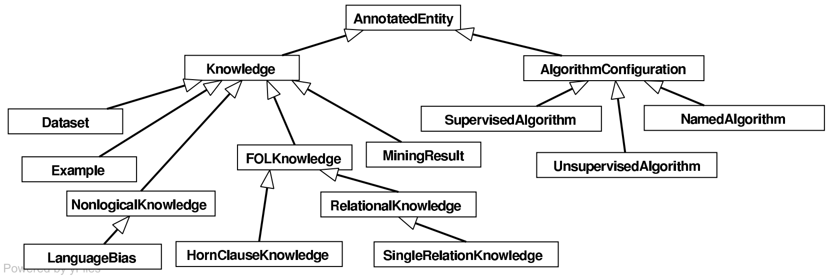

Fig. 1. The basic top level structure of the knowledge discovery ontology.

mining algorithms. It focuses on classical data mining processes, which contain three subsequent steps: pre-processing, model induction and post-processing, while our primary focus is in describing more complex relational data mining tasks. The workflows generated by the NExT system are linear, whereas for our tasks workflow is a directed acyclic graph.

The development of a unified theory (conceptualization) of data mining was recently identified as the first of ten most challenging problems for data mining research [19]. While we do not claim completeness or universal applicability of the ontology developed in this work, in its design we did try to follow the state-of-the-art works attempting to establish such a unified theory. From Mannila's traditional definition of data mining [9], we accepted the core concepts of a pattern and representation language. On the other hand, in categorizing knowledge and algorithm types, we followed on the recent comprehensive study by Džeroski [6].

3 Knowledge Discovery Ontology

Our knowledge discovery ontology defines relationships among diverse ingredients of knowledge discovery scenarios, including both declarative (various knowledge representations) and algorithmic (both inductive and deductive algorithms). The primary purpose of the ontology is to make the workflow planner able to reason about which algorithms can be used to produce intermediary or final results required by a specified data mining task. Due to limited space, we constrain ourselves to describing only the basic aspects of our approach to designing the ontology, which essentially follows up on the recent attempts of establishing a conceptual framework for data mining [6]. Our proposal addresses two core concepts: *knowledge*, capturing the declarative elements in knowledge discovery, and *algorithms*, which serve to transform a piece of knowledge into another piece of knowledge. The basic top-level structure of the ontology is in Figure 1. Currently the ontology contains about 70 classes including descriptions of some propositional algorithms available in Weka data mining platform [18] and relational data mining algorithms described in [15].

As an example of algorithm description we present the definition of the well known Apriori algorithm in the description logic notation [1] :

$$
\begin{aligned}
\texttt{Apriori} \sqsubseteq\; & \texttt{NamedAlgorithm} \\
& \sqcap \exists\, \texttt{output} \cdot (\texttt{MiningResult} \sqcap \\
& \quad \forall\, \texttt{contains} \cdot \texttt{AssociationRule}) \\
& \sqcap \exists\, \texttt{input} \cdot (\texttt{Dataset} \sqcap \\
& \quad \texttt{SingleRelationKnowledge} \sqcap \\
& \quad \exists\, \texttt{format} \cdot \{\texttt{ARFF}, \texttt{CSV}\}) \\
& \sqcap \exists\, \texttt{minSupport} \cdot \mathit{double} \\
& \sqcap \exists\, \texttt{minConfidence} \cdot \mathit{double}
\end{aligned}
$$

The Apriori algorithm is defined as an algorithm that has two parameters `minSupport` and `minConfidence`, has a single relation dataset in the CSV or ARFF format as its input, and produces a result in the form of association rules.

Technically, the ontology is implemented in the description logic variant (OWL-DL) of the leading semantic web language OWL [11]. Our primary reasons for this choice were OWL's sufficient expressiveness, modularity, availability of ontology authoring tools and optimized reasoners and a well-established community support.

4 Workflow Construction

The task of automatic workflow construction consists of the following steps: converting the KD task into a planning problem, generating the plan using a third party planning algorithm, storing the generated abstract workflow in form of semantic annotation, instantiating the abstract workflow with specific configurations of the required algorithms and storing the generated workflow.

To maintain generality of our approach, we decided to encode the planning task into the standard language PDDL ('Planning Domain Definition Language') [12]. We are using PDDL 2.0 with type hierarchy and domain axioms. Planning algorithms require two main inputs. The first one is a description of the domain specifying the available types of objects and actions. The second one is the problem description specifying initial state, goal state and the available objects. We have developed an algorithm for generating the domain description from the KD ontology. In order to formalize problem description and generate the problem description in PDDL in a similar way and for storing the created workflows in a knowledge-based representation, we have created a small ontology for workflows, which extends the knowledge discovery ontology.

As an example we present the definition of action in PDDL representing the Apriori algorithm described in Section 3.

```
(:action AprioriAlgorithm
 :parameters (
  ?v0 - Dataset_SingleRelationKnowledge
  ?v1 - CSV
  ?v2 - MiningResult_contains_Associa-
        tionRule )
 :precondition (and (available ?v0)
                    (format ?v0 ?v1))
 :effect (and (available ?v2)
              (format ?v2 ?v1)))
```

The information about the output of Apriori algorithm was expressed using a conjunction of the named ontological class *MiningResult*, a universal restriction on *contains* and an existential restriction on *format*. Therefore the effects of the action using Apriori algorithm are represented using the unary predicate *available* applied on a named class *MiningResult_contains_AssociationRule*, which is a subclass of *MiningResult*, and a binary predicate *format*.

We have implemented a planning algorithm based on the Fast-Forward (FF) planning system [7] to generate abstract workflows automatically. The FF planning system uses a modified version of hill climbing algorithm called enforced hill climbing to perform forward state space search. The goal distances are estimated by relaxed GRAPHPLAN [3]. The original planning problem is converted into a relaxed problem by ignoring delete lists of the operators.

Currently the planning algorithm outputs the first workflow with the smallest number of processing steps as the solution. In future work we are planning to include other heuristics such as the estimated runtimes of the workflows to provide the user with the possibility to view and select from a number of workflows with the highest ranking.

5 Use Cases

We have conducted experiments with workflow construction in two domains. The first domain is genomics, where we were interested in relational descriptive analysis of gene expression data. The second is concerned with learning from product design data. Here the examples are semantically annotated CAD documents.

Both these domains are highly knowledge-intensive. One of the main challenges is to efficiently extract relevant information from large amounts of data from different sources with a rich relational structure. The use of advanced knowledge engineering techniques is becoming popular not only in bioinformatics, but also in the engineering domain, and complex background knowledge thus nowadays characterizes both domains. As a result, traditional data mining techniques and tools are not straightforwardly applicable. Rather, complex knowledge discovery workflows are required in both the domains under inquiry.

An example of abstract workflow generated in the engineering domain is in Figure 2. There are four preprocessing steps, which can be performed simultaneously. In this case all the preprocessing steps focus on extracting knowledge

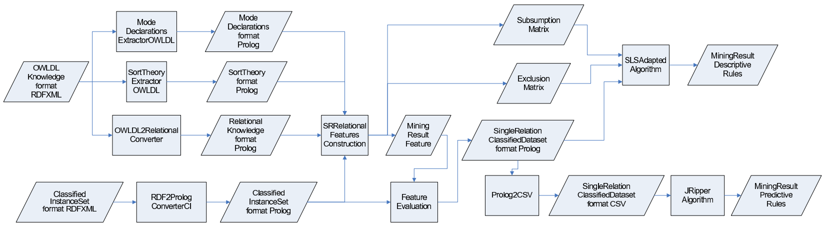
OWLDL Knowledge format RDFXML
Mode Declarations ExtractorOWLDL
SortTheory Extractor OWLDL
OWLDL2Relational Converter
Mode Declarations format Prolog
SortTheory format Prolog
Relational Knowledge format Prolog
SRRelational Features Construction
Classified InstanceSet format RDFXML
RDF2Prolog ConverterCI
Classified InstanceSet format Prolog
Subsumption Matrix
Exclusion Matrix
Mining Result Feature
SingleRelation ClassifiedDataset format Prolog
Feature Evaluation
SLSAdapted Algorithm
MiningResult Descriptive Rules
Prolog2CSV
SingleRelation ClassifiedDataset format CSV
JRipper Algorithm
MiningResult Predictive Rules

Fig. 2. An automatically generated workflow for obtaining classification (predictive) rules and subgroup descriptions (descriptive rules) from annotations of CAD design drawings.

from the semantic representation into a form, in which it could be used by relational data mining algorithm. *ModeDeclarationsExtractorOWLDL* extracts mode declarations from domain and range restrictions on properties defined in the CAD ontology. The sort theory containing a taxonomy of classes from the CAD ontology is extracted using *SortTheoryExtractorOWLDL*. The *OWLDL-RelationalConverter* is used to convert descriptions of the individual annotations to Prolog and *RDF2PrologConverterCI* does the same for the identifiers of the annotations.

6 Conclusions and Future Work

We entered this study with the primary hypothesis that complex scientific and engineering knowledge discovery workflows, such as those we had to develop manually in previous studies [14, 16], can be proposed semi-automatically. Semi-automatic workflow composition does require the user to know exactly what he/she possesses as the knowledge input and what kind of output he/she desires to achieve, but it does not require him/her to be aware of the numerous properties and mutual relationships of the wide range of relevant knowledge discovery algorithms.

For the purpose of workflow generation, we used two main ingredients. First, a formal conceptualization of knowledge types and algorithms was implemented through a *knowledge discovery ontology*, following up on state-of-the-art developments of a unified data mining theory. Second, a planning algorithm is employed that assembles workflows on the basis of planning task descriptions extracted from the knowledge discovery ontology and the given user's input-output task requirements. The workflows generated by our algorithm were complex, but reasonable in that there was no apparent way of simplifying them while maintaining the desired functionality. Therefore the workflows generated in two use cases (in science and engineering) can serve as a proof of concept for our approach.

Since the generated workflows are not linear, we could get significant runtime improvements from executing the workflows in the GRID environment. Therefore in future work we are planning to extend the ontology by descriptions of concrete computational resources e.g. by integration of our KD ontology into OWL-S. This will enable us to produce workflows optimized for execution in a given computing environment. We are also planning to evaluate the respective merits of planning via conversion into PDDL and building a planning algorithm directly over the DL representation.

Acknowledgements

This research was supported by the SEVENPRO, IST-027473 project in FP6 and by the 1ET101210513 project ML/BIO: Relational machine learning for analysis of biomedical data.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook, Theory, Implementation and Applications.* Cambridge University Press, 2003.
2. A. Bernstein and M. Deanzer. The next system: Towards true dynamic adaptions of semantic web service compositions (system description). In *Proceedings of the 4th European Semantic Web Conference (ESWC'07).* Springer, 2007.
3. A. Blum and M. Furst. Fast planning through planning graph analysis. *Artificial intelligence*, 90:281–300, 1997.
4. P. Brezany, I. Janciak, and A. M. Tjoa. Ontology-based construction of grid data mining workflows. In H.O. Nigro, S.E.G. Cisaro, and D.H. Xodo, editors, *Data Mining with Ontologies: Implementations, Findings and Frameworks.* IGI Global, 2007.
5. E. Deelman, J. Blythe, Y. Gill, C. Kesselman, S. Koranda, A. Lazzarini, G. Mehta, M.A. Papa, and K. Vahi. Pegasus and the pulsar search: From metadata to execution on the grid. In *Parallel Processing and Applied Mathematics.* Springer, 1990.
6. S. Dzeroski. Towards a general framework for data mining. In S. Dzeroski and J. Struyf, editors, *Knowledge Discovery in Inductive Databases - 5th International Workshop, KDID'06, Revised, Selected and Invited Papers*, volume 4747, pages 259–300, 2007.
7. J. Hoffmann and B. Nebel. Online convex programming and generalized infinitesimal gradient ascent. *Journal of Artificial Intelligence research*, 14:253–302, 2001.
8. M. Klusch, A. Gerber, and M. Schmidt. Semantic web service composition planning with owls-xplan. In *1st Intl. AAAI Fall Symposium on Agents and the Semantic Web*, 2005.
9. H. Mannila. Aspects of data mining. In *Procs of the MLnet Familiarization Workshop on Statistics, Machine Learning and Knowledge Discovery in Databases*, 1995.
10. K. Morik and M. Scholz. The miningmart approach to knowledge discovery in databases. In Intelligent Technologies for Information Analysis, editor, *Proceedings of the International Conference on Machine Learning*, pages 47–65. Springer, 2004.
11. P. Patel-Schneider, P. Hayes, and I. Horrocks. Owl web ontology language semantics and abstract syntax, 2004.

12. D. Smith and D. Weld. Temporal planning with mutual exclusion reasoning. In *Proceedings of the 1999 International Joint Conference on Artificial Intelligence (IJCAI-1999)*, pages 326–333, 1999.
13. A. Suyama, N. Negishi, and T. Yamagchi. Composing inductive applications using ontologies for machine learning. In S. Arikawa and H. Motoda, editors, *Proceedings of the First International Conference on Discovery Science*, pages 429 – 431, 1998.
14. I. Trajkovski, F. Zelezny, N. Lavrac, and J. Tolar. Learning relational descriptions of differentially expressed gene groups. *IEEE Trans. Sys Man Cyb C*, 38(1):16–25, 2008.
15. M. Žáková and F. Železný. Exploiting term, predicate, and feature taxonomies in propositionalization and propositional rule learning. In *ECML 2007: 18th European Conference on Machine Learning*, 2007.
16. M. Žaková, F. Železný, J. A. Garcia-Sedano, C. Massia-Tissot, N. Lavrač, P. Křemen, and J. Molina. Relational data mining applied to virtual engineering of product designs. In *Procs of the 16th Int. Conference on Inductive Logic Programming*, volume 4455, pages 439–453. Springer, 2006.
17. R. Wirth, C. Shearer, U. Grimmer, T. P. Reinartz, J. Schloesser, C. Breitner, R. Engels, and G. Lindner. Towards process-oriented tool support for knowledge discovery in databases. In *Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery*, volume 1263, pages 243 – 253, 1997.
18. I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, 2005.
19. Q. Yang and X. Wu. 10 challenging problems in data mining research. *Intl. Jrnl. of Information Technology and Decision Making*, 5(4):597–604, 2006.