Designing KDD-Workflows via HTN-Planning

Kietz, Jörg-Uwe ; Serban, Floarea ; Bernstein, Abraham ; Fischer, Simon

DOI: https://doi.org/10.3233/978-1-61499-098-7-1011

Originally published at:
Kietz, Jörg-Uwe; Serban, Floarea; Bernstein, Abraham; Fischer, Simon (2012). Designing KDD-Workflows via HTN-Planning. In: European Conference on Artificial Intelligence, Systems Demos, Montpellier, France, 27 August 2012 - 31 August 2012. I O S Press, 1011-1012.
DOI: https://doi.org/10.3233/978-1-61499-098-7-1011

ECAI 2012
Luc De Raedt et al. (Eds.)

doi:10.3233/978-1-61499-098-7-1011

Designing KDD-Workflows via HTN-Planning

Jörg-Uwe Kietz[1] and **Floarea Serban**[1] and **Abraham Bernstein**[1] and **Simon Fischer**[2]

Abstract. Knowledge Discovery in Databases (KDD) has evolved a lot during the last years and reached a mature stage offering plenty of operators to solve complex data analysis tasks. However, the user support for building workflows has not progressed accordingly. The large number of operators currently available in KDD systems makes it difficult for users to successfully analyze data. In addition, the correctness of workflows is not checked before execution.

This demo presents our tools, eProPlan and eIDA, which solve the above problems by supporting the whole cycle of (semi-) automatic workflow generation. Our modeling tool eProPlan, allows to describe operators and build a task/method decomposition grammar to specify the desired workflows. Additionally, our Intelligent Discovery Assistant, eIDA, allows to place workflows into data mining (DM) suites or workflow engines for execution.

1 Introduction

One of the challenges of KDD is assisting users in creating and executing workflows. Existing KDD systems such as the commercial IBM SPSS Modeler[3] or the open-source RapidMiner [4] support the user with nice graphical user interfaces. Operators can be dropped as nodes onto the working pane and the data-flow is specified by connecting the operator-nodes. This works very well as long as neither the workflow becomes too complicated nor the number of operators becomes too large. In the past decade, however, the *number of operators* in such systems has been growing fast. All of them contain over 100 operators and RapidMiner (RM)—a popular open source KDD system—now has around 1000. In addition to the number of operators also the *workflows' size* has been growing in recent years. Today's workflows easily contain hundreds of operators. Parts of the workflows are applied several times implying that the users either need to copy/paste or even repeatedly design the same sub-workflow. Furthermore, workflows are not checked for *correctness* before execution: the execution frequently stops with an error after running for several hours due to small syntactic incompatibilities between an operator and the data it should be applied on.

To address these problems several authors [1, 3, 9] propose the use of planning techniques to automatically build workflows. All these approaches are, however, limited in several ways. First, they only model a small set of operations and are working only for short workflows (less than 10 operators). Second, none of them model operations that work on individual attributes of a data set: they only model operations that process all attributes of a data set equally together. Lastly, the approaches cannot scale to large amounts of operators and large workflows as their planning approaches fail in the large design space of "correct" solutions. A full literature review about IDAs (including these approaches) can be found in our survey [7].

In this paper we describe the first approach for designing KDD workflows based on ontologies and Hierarchical Task Network (HTN) planning [4]. *Hierarchical task decomposition knowledge* available in DM (e.g. CRISP-DM [2]) can be used to significantly reduce the number of generated unwanted correct workflows. Thus, KDD researchers can easily model not only their DM and preprocessing operators but also their DM tasks that are used to guide the workflow generation. Moreover less experienced users can use our RM-IDA plugin to automatically generate workflows in only 7 clicks.

2 The Overall System

Our system has two main components as illustrated in Fig. 1: eProPlan our modeling support tool for new operators and new tasks to be solved by the planner and eIDA which generates and deploys workflows into DM-suites. eProPlan is the modeling environment for the DMWF ontology, which describes the KDD domain. It allows to model new operators and uses a task-method decomposition grammar to solve DM problems. Designed as a plugin for the open-source ontology-editor Protégé 4[5], eProPlan exploits the advantages of the ontology as a formal model for the domain knowledge. Instead of over-using the ontological inferences for planning (as in [3, 9]) we extend the ontological formalism with the main components of a plan, namely operator conditions & effects for classical planning and tasks-methods decomposition grammar for HTN-planning. The planner is implemented in Flora2/XSB [8] and uses the DMWF ontology as a planning domain [6]. The planning problem consists of the meta-data of the data set and a set of goals/hints entered by the user.

eIDA is a programming interface to the reasoner & planner used to plugin an IDA into existing systems (so far RapidMiner and Taverna[7] rely on it). For a given dataset it allows to retrieve the plans by passing its meta-data and the main DM goal. More detailed papers,

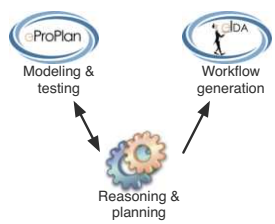

Figure 1: The eProPlan architecture

[1] University of Zurich, Department of Informatics,Dynamic and Distributed Information Systems Group,Binzmühlestrasse 14, CH-8050 Zurich, Switzerland {kietz|serban|bernstein}@ifi.uzh.ch
[2] Rapid-I GmbH, Stockumer Str. 475, 44227 Dortmund, Germany fischer@rapid-i.com
[3] http://www.ibm.com/software/analytics/spss/
[4] http://rapid-i.com/content/view/181/190/
[5] http://protege.stanford.edu/
[6] Traditionally, planners only find the first solution whereas our problem is unconstrained: the first encountered solution is usually not the best one.
[7] http://www.taverna.org.uk/

the demo video and all software[8] described here are freely available and linked for download from http://www.e-lico.eu/.

3 Demonstration

The demonstration has two steps:

First, it presents the generation of complete workflows via the RM-IDA in only "7 clicks" (see Figure 2). (1) Go to the IDA-Perspective; (2) drag the data to be analyzed from the repository to the view or import (and annotate) your data; (3) select the main DM goal; (4) ask the IDA to generate workflows; (5) evaluate all plans by executing them in RM; (6) select the plan you like most to see its summary (the screenshot in Figure 2 was made after this step); and finally (7) inspect the plan and its results. Without the IDA, DM is only achievable by specialized highly-trained professionals. They need knowledge about DM methods and how they are implemented in RapidMiner. They have to inspect the data and combine these operators into an adequate workflow. The RM-IDA does not require detailed technical knowledge anymore. The user still needs some knowledge about DM, i.e. the statistical assumptions underlying DM. But this is knowledge required in any experimental science.

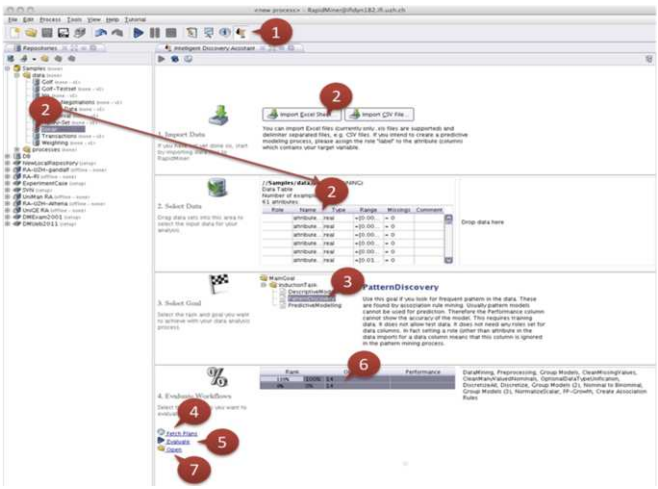

Figure 2: IDA Interface in RapidMiner

Second, it shows the modeling of data sets, DM-operators, DM-goals and task/method decompositions, via testing the model in eProPlan by entering specific goals and getting the DMWF-meta-data description of concrete data sets from a data analysis service.

4 Evaluation of the IDA

We tested the IDA on 108 data sets from the UCI repository [9]. It produced executable plans for 78 classification and 30 regression problems. These data sets have between 3 and 1558 attributes, that are all nominal, all scalar (normalized or not), or mixed types. They have varying degrees of missing values. Note, that we are not aware of another Machine Learning or DM approach that can adapt itself to so many different and divergent data sets. The IDA also works well for less prepared data sets like the KDD Cup 1998 challenge data, where it generates plans with 40 operators. Generating and ranking 20 of these workflows took 400 sec. on a 3.2 GHz Quad-Core Intel Xeon.

Besides making DM easier for inexperienced users, an additional goal for building the IDA was to speed-up the design of DM workflows. To this end we compared the performance of computer science students at the end of a DM class to the results gained by using an IDA by a non-specialist when solving standard DM problems (such as clustering and prediction tasks on two complex UCI data sets). The study confirmed that the IDA was faster to attain a comparable quality: the students solved the tasks in 3 hours; the IDA did it in 30 minutes.

The planner was evaluated by our project partners who used probabilistic ranking and meta-mining [5, 6]. Their evaluation was done on 65 high dimensional biological datasets with few instances/samples. For their experiments they cross-validated all performance by holding out a dataset. The resulting meta-model was then used to rank the IDA-generated workflows. They found that the meta-learned rankings significantly outperformed the default, frequency-based strategy. Hence, their ranker was able to improve on our ranking to find DM workflows that maximize predictive performance.

5 Conclusions

We presented our Intelligent Discovery Assistant (eIDA and eProPlan) for planning KDD workflows. eIDA can be easily integrated into existing DM-suites or workflow engines. eProPlan is a user-friendly environment for modeling DM operators. Furthermore, it is able to plan attribute-wise operations. The main scientific contribution of this IDA demonstration is its ability to build complex workflows out of a much larger set of operations than all previous systems. The demo presents how planning-based KDD workflow design can significantly help KDD practitioners to make their daily work more efficient.

ACKNOWLEDGEMENTS

This work is partially supported by the European Community 7^{th} framework program ICT-2007.4.4 under grant number 231519 "e-Lico: An e-Laboratory for Interdisciplinary Collaborative Research in Data Mining and Data-Intensive Science".

REFERENCES

[1] Abraham Bernstein, Foster Provost, and Shawndra Hill, 'Towards Intelligent Assistance for a Data Mining Process: An Ontology-based Approach for Cost-sensitive Classification', *IEEE Transactions on Knowledge and Data Engineering*, **17**(4), 503–518, (April 2005).

[2] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, 'Crisp–dm 1.0: Step-by-step data mining guide', Technical report, The CRISP–DM Consortium, (2000).

[3] C. Diamantini, D. Potena, and E. Storti, 'KDDONTO: An Ontology for Discovery and Composition of KDD Algorithms', in *Proceedings of the SoKD-09 Workshop at ECML/PKDD09*, (2009).

[4] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: Theory & Practice*, Morgan Kaufmann, San Francisco, CA, USA, 2004.

[5] P. Nguyen, A. Kalousis, and M. Hilario, 'A meta-mining infrastructure to support kd workflow optimization', in *Proc. of the PlanSoKD-2011 Workshop at ECML/PKDD-2011*, (2011).

[6] Phong Nguyen and Alexandros Kalousis, 'Evaluation report on meta-miner'. Deliverable 7.2 of the EU-Project e-LICO, January 2012.

[7] F. Serban, J. Vanschoren, J.-U. Kietz, and A. Bernstein, 'A Survey of Intelligent Assistants for Data Analysis', *ACM Computing Surveys*, (to appear 2012).

[8] G. Yang, M. Kifer, and C. Zhao, 'Flora-2: A Rule-Based Knowledge Representation and Inference Infrastructure for the Semantic Web', *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, 671–688, (2003).

[9] M. Žaková, P. Křemen, F. Železný, and N. Lavrač, 'Automating knowledge discovery workflow composition through ontology-based planning', *Automation Science and Engineering, IEEE Transactions on*, **8**(2), 253 –264, (2011).

[8] The RM-IDA extension can be auto-installed from inside RapidMiner by switching the update-server to http://rapidupdate.de:8180/UpdateServer.

[9] http://archive.ics.uci.edu/ml/datasets.html