Auto-experimentation of KDD Workflows Based on Ontological Planning

Floarea Serban

University of Zurich, Department of Informatics,
Dynamic and Distributed Information Systems Group,
Binzmühlestrasse 14, CH-8050 Zurich, Switzerland
serban@ifi.uzh.ch

Abstract. One of the problems of Knowledge Discovery in Databases (KDD) is the lack of user support for solving KDD problems. Current Data Mining (DM) systems enable the user to manually design workflows but this becomes difficult when there are too many operators to choose from or the workflow's size is too large. Therefore we propose to use auto-experimentation based on ontological planning to provide the users with automatic generated workflows as well as rankings for workflows based on several criteria (execution time, accuracy, etc.). Moreover auto-experimentation will help to validate the generated workflows and to prune and reduce their number. Furthermore we will use mixed-initiative planning to allow the users to set parameters and criteria to limit the planning search space as well as to guide the planner towards better workflows.

1 Introduction

The technology advances in the last decade facilitate the generation of large amount of data. One of today's problems is how to process and extract patterns from it. Knowledge Discovery in Databases (KDD) has made progress during the last years, since the new types of data that appeared (text, image, multimedia) generated new algorithms to handle them. Therefore current KDD systems incorporate more and more operators. But this creates problems for users since they are confronted with a plethora of operators. Hence it becomes difficult to figure out the best choice from so many options (operators).

One of the main issues of such systems is the level of user support. KDD systems like Weka[1], RapidMiner[2], KNIME[3], EnterpriseMiner[4] or Clementine[5] provide the user nice graphical interfaces that allow them to design KDD workflows. Users can drag and drop operators and connect them. Also explanations about operators' functionalities and parameters are supported.

[1] http://www.cs.waikato.ac.nz/ml/weka/
[2] http://rapid-i.com/content/view/181/190/
[3] http://www.knime.org/
[4] http://www.sas.com/technologies/analytics/datamining/miner/
[5] http://www.spss.com/software/modeling/modeler-pro/

P.F. Patel-Schneider et al.(Eds.): ISWC 2010, Part II, LNCS 6497, pp. 313–320, 2010.

However taking into account the large number of operators the existing support does not help the users solve their tasks. It becomes cumbersome to select the right operator as well as to build the right workflow. One of the solutions proposed by several authors is to use AI planning techniques to automatically generate workflows [1,23,7,22]. But current implementations are limited since they support a reduced number of operators (not more than 50) as well as the generated workflows contain not more than 10 operators. The problem is that this is not enough to successfully solve Data Mining (DM) tasks.

To overcome these problems we came with the idea of using an ontology to model the DM domain that incorporates more operators [15] than the other ontologies, and use AI planning to automatically generate plans. Opposed to other approaches we use Hierarchical Task Planning (HTN) [11] since hierarchical task decomposition (from CITRUS [22]) and the knowledge available in the DM domain (CRISP-DM standard [6]) can significantly reduce the number of generated workflows. Indeed this limits the number of unwanted workflows but it is still hard for a user to decide which workflows to choose and execute. Other approaches use meta-learning to find out which operators are best for a specific data set [13,3], focusing more on classification algorithms, without a striking outcome.

Therefore we propose to use systematic auto-experimentation of generated plans to discover heuristics that prune the number of generated workflows as well as structure them. By automatic experimentation we mean running the plans retrieved from the planner in order to find the best plans or a ranking for the overall plans. Our approach will run all the plans for different data sets and try to learn from the results of the experiments. The auto-experimentation will not only improve the ranking of plans, but will also evaluate the outcome of the generated plans, acting as a validation module for the planning system. Furthermore, we propose a mixed-initiative planning approach where the user can define hints or criteria to prune the planning search process and guide the planner towards better workflows.

The remainder of the paper is structured as follows: The next section describes the current state of related research, Section 3 discusses the current state of the work and presents the future research steps. The paper closes with conclusions in Section 4.

2 Related Work

Several approaches try to improve user support for DM by providing intelligent assistance [21]. However all these approaches are either a proof of concept or limited to a small number of operators, therefore they are not usable for todays DM tasks. Existing IDAs do not include any support for auto-experimentation except maybe with the attempt of [1] which enable the user to execute workflows based on a ranking. Yet they don't improve the generation of workflows based on the previous executions.

Planning and ontologies. Considering planning and ontologies to automatically generate KDD workflows has been suggested and tried by several researchers.

They use ontologies to organize and structure the DM domain and then they use it for planning [1,7,24]. But most of the existing ontologies are rather simple, except for [23,24] which include a larger number of operators (more than 60). But even so they are not able to plan very large workflows. IDEA [1] offers an IDA that uses a prototype ontology and a DM-process planner to systematically enumerate and rank DM processes by speed and accuracy. However both the IDA and the ontology are prototypes and limited to a specific number of DM operators.

The framework presented in [24] automatically constructs DM workflows based on input and output specification of the data mining task based on a Knowledge Discovery ontology which is used for planning DM workflows. Their approach uses a Fast-Forward algorithm for planning combined with the benefits of the hierarchy from the ontology.

The approach proposed by [8] uses a KDD ontology to support KDD process design. Moreover they use a semantic matching function for the automatic composition of algorithms forming valid prototype KDD processes.

We choose to base our system on the approach of [15]. They built a DM ontology (developed within the e-Lico project[6]) which contains more than 70 operators as well as conditions and effects for each operator expressed in an extended SWRL language[7] with a set of needed built-ins. Moreover they use HTN planning to be able to reduce the number of generated workflows, but also because HTN planning proved to be better in solving real problems [20].

Auto-experimentation of KDD workflows. A recent discovered approach is the one of [9] which try to improve the execution of KDD workflows generated by AI planners. We can say this is the closest work to our approach. They propose a distributed architecture for automating the KDD processes as well they include a learning module which can learn from the execution of previous workflows. But they focus only on classification and regression as well they separate the execution of pre-processing actions. Their work is close to meta-learning. Our work will focus on the whole workflow and try to find different metrics to evaluate its performance. Similar to their approach we are also going to define different quality criteria which the user can set before executing the plans such that we limit the auto-experimentation. But our approach will focus on generating hypotheses based on the plans execution, which are better plans.

Meta-learning. Experimental databases are proposed by [2] to store Machine Learning (ML) experiments. They facilitate large-scale experimentation, guarantee repeatability of experiments and improve reusability of experiments. They also use meta-learning to determine the most appropriate ML tool for a data set. But their are focusing on a single step of the DM process, in fact on a ML tool (or algorithm).

MetaL uses the notion of meta-learning to advise users which induction algorithm to choose for a particular data-mining task [13]. One of the outcomes

[6] `http://www.e-lico.eu/`
[7] `http://www.w3.org/Submission/SWRL/`

of the project was a Data Mining Advisor (DMA) [12] based on meta-learning that gives users support with model selection. IDM has a knowledge module which contains meta-knowledge about the data mining methods, and it is used to determine which algorithm should be executed for a current problem.

Other work was done by the StatLog project [16] which has investigated which induction algorithms to use given particular circumstances. This approach is further explored by [3,10] which use meta-rules drawn from experimental studies, to help predict the applicability of different algorithms; the rules consider measurable characteristics of the data (e.g., number of examples, number of attributes, number of classes, kurtosis, etc.). [4] present a framework which generates a ranking of classification algorithms based on instance based learning and meta-learning on accuracy and time results. However, all these approaches are studying the execution of only one DM algorithm. Our approach will focus on entire DM workflow.

Mixed-initiative planning. Several approaches proposed different techniques to involve the user in the planning process, which are usually known under the name of mixed-initiative or collaborative planning. One of the approaches is the one described in [18] and known as *advisable planning* which attempts to model the behavior of the planner before starting the planning process. Our ontology already contains such mixed-initiative planning facility and based on it the user will be able to refine the planning goal and its steps. Another approach is *configurable planning* suggested by [19] which is a combination of domain-independent planning engines with higher-level abstractions like HTNs that capture and exploit domain knowledge. Since our approach is based on HTNs we are already using such domain knowledge.

3 Research Plan

We propose to integrate auto-experimentation of DM workflows, generated using ontological planning, in existing IDAs. The automatic experimentation approach will provide heuristics to simplify and improve the planning process as well as rank the plans according to different metrics such as accuracy, length of workflows, execution time, etc.

The main purpose of the system is to assist users in the generation and experimentation of DM workflows, as well as guiding them to configure parameters for achieving best performance. The target group consists of KDD researchers and people who are familiar with DM terminology. Later on we will try to extend it for naive users.

3.1 Current State

Ontological planning. We choose to base our approach on ontological planning [14] since the ontology offers a hierarchical structure of the DM concepts. Moreover it enables us to define conditions and effects for operators as SWRL rules which are essential for planning. The planner uses a DM ontology as a planning domain. To be able to use the ontology for planning we need to compile

it in a format that the planner can understand. So far we have been involved in developing the compilation of the DM ontology (the DMWF - Data Mining Workflow ontology [14]) such that the planner can use it as a domain for generating plans. The compilation consists of compiling the TBox (terminology which does not include annotations), then the operators together with their conditions and effects (operators are classes but their conditions and effects are stored as annotations and we need to compile them separately), inputs, outputs and parameters as well as the task/method decomposition (the same goes for the task and method decomposition, the structure is saved as annotations as well). Finally we compile the ABox.

Experiments' design. Based on the ontology and planner we can start designing experiments. The main task was to implement an IDA-API which is able to retrieve plans starting from a goal definition and a provided data set. The IDA-API has the following features:

- The DM task can be specified in the form of main goals and optional goals.
- The meta data from the used data sets can be added as a list of facts.
- The plans can be easily retrieved.

The architecture of our system can be seen in Figure 1. We are using the IDA-API to define the DM problems and to retrieve the generated plans. The developed IDA-API uses a pre-compiled DM ontology (as described before) that is later used for planning. Then it compiles the task definition and the meta-data of the data sets used as inputs into a set of facts that can be recognized by the planning module.

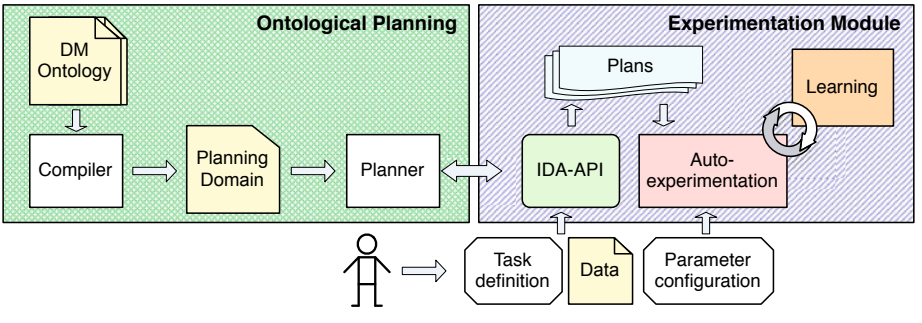

Fig. 1. System architecture

3.2 Plan for Future Research

Having laid the foundations for auto-experimentation of KDD workflows, we can start working on the future steps to achieve our contributions described in Section 1 as follows:

Auto-experimentation to discover heuristics to prune the search plan. Having a large set of plans to choose from is a challenge for the experimentation module. In this part of the project we develop an experimentation module

which is able to automatically run experiments in an optimal way. Moreover the experiments need to be analyzed and used for learning and improving the experimentation as well as the planning process.

Firstly we need to decide which DM system to use to run the experiments. We incline towards RapidMiner [17], since it is one of the leading open source DM tools. Moreover RapidMiner is implemented in Java which makes it easy to be used with our IDA-API.

Secondly, we need to implement a module that based on the IDA-API enables us to easily define DM tasks. A simple approach would be to have that programatically but later on we need to develop a graphical interface for it.

Thirdly, we need to build a system that enables us to run DM experiments in a distributed manner. For starting, the plans will be executed in parallel, one plan in a thread. Later on we can find ways of parallelizing sub-workflows.

Also we need to study the results of the experiments and find good heuristics to reduce the number of generated plans. Then we need to evaluate the metrics and heuristics by performing several types of tasks on a large number of data sets. For that we will develop a learning module which analyzes the results of the experiments and tries to improve the auto-experimentation. As a starting point, we could use the data sets from the UCI repository[8] and later test it on larger data sets (also which are not preprocessed). Another idea would be to generate hypotheses (which would represent better plans) based on the experiments and then run them and check their accuracy. In the end, our purpose is to be able to solve one of the KDDCups[9] (for example KDDCup'98) using the system and show that the auto-experimentation module can successfully provide a reduced number of plans and qualitative rankings.

Mixed-initiative planning. There are many parameters that can influence and improve the results of the experiments, for example, the time of the experiments, the resources used, the accuracy, etc.. The main challenge is to find the best set of parameters which can lead to significant improvement of the experimentation module. Another one is to provide the user the possibility to configure the experiments and to influence the planning process.

Firstly, we need to find a set of qualitative metrics the user could set to improve and guide the planning search. Secondly, we will design a GUI the allows the user to set all these parameters. Then, we will allow the user to visualize and manipulate plans by integrating actions like plan step by step, pause or execute a plan, go next or go back one step. We will later try to extend this approach and allow the user to contribute not only to the formulation and development of plans, but also in the management, refinement, analysis and repair of the plans. But first we need to study and analyze all the problems raised in [5].

Finally, we will perform user tests and check if the generated system helps the users to solve their tasks better and faster than the existing DM systems.

[8] `http://archive.ics.uci.edu/ml/`
[9] `http://www.sigkdd.org/kddcup/index.php`

4 Conclusions

In this paper we introduce auto-experimentation of KDD workflows based on ontological planning. We extend upon research described in Section 2 in various ways. Firstly, we use auto-experimentation to reduce and prune the number of automatically generated workflows. Secondly, we integrate the auto-experimentation module into an IDA and allow the users to browse workflows by rankings and analyze the outcomes of their execution. Thirdly, we will provide a mixed-initiative module that allows the users to guide the planning process as well as to suggest criteria to prune the searching space.

We are optimistic and believe that the current approach will lead to different ways of ranking and structuring of the plans as well as involve the users in the planning process. The impact of our approach is the possibility to find rankings for DM workflows and heuristics to prune the planner searching space, hence reducing the time needed to generate plans and finding the best workflow for a specific DM problem through auto-experimentation.

Acknowledgements. This work is supported by the European Community 7^{th} framework ICT-2007.4.4 (No 231519) "e-Lico: An e-Laboratory for Interdisciplinary Collaborative Research in Data Mining and Data-Intensive Science".

References

1. Bernstein, A., Provost, F., Hill, S.: 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 (2005)
2. Blockeel, H., Vanschoren, J.: Experiment databases: Towards an improved experimental methodology in machine learning. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 6–17. Springer, Heidelberg (2007)
3. Brazdil, P., Gama, J., Henery, B.: Characterizing the applicability of classification algorithms using meta-level learning. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 83–102. Springer, Heidelberg (1994)
4. Brazdil, P., Soares, C., Da Costa, J.: Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. Machine Learning 50(3), 251–277 (2003)
5. Burstein, M., McDermott, D.: Issues in the development of human-computer mixed-initiative planning. Advances in Psychology 113, 285–303 (1996)
6. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: Crisp–dm 1.0: Step-by-step data mining guide. Technical report, The CRISP–DM Consortium (2000)
7. Diamantini, C., Potena, D., Storti, E.: Kddonto: An ontology for discovery and composition of kdd algorithms. In: Service-oriented Knowledge Discovery (SoKD 2009) Workshop at ECML/PKDD 2009 (2009)
8. Diamantini, C., Potena, D., Storti, E.: Supporting users in kdd processes design: a semantic similarity matching approach. In: Planning to Learn Workshop (PlanLearn 2010) at ECAI 2010, pp. 27–34 (2010)

9. Fernández, S., Súarez, R., de la Rosa, T., Ortiz, J., Fernández, F., Borrajo, D., Manzano, D.: Improving the execution of kdd workflows generated by ai planners. In: Planning to Learn Workshop (PlanLearn 2010) at ECAI 2010, pp. 19–25 (2010)
10. Gama, J., Brazdil, P.: Characterization of classification algorithms. In: Progress in Artificial Intelligence, pp. 189–200 (1995)
11. Ghallab, M., Nau, D., Traverso, P.: Automated Planning: Theory & Practice. Morgan Kaufmann, San Francisco (2004)
12. Giraud-Carrier, C.: The data mining advisor: meta-learning at the service of practitioners. In: Proceedings of Fourth International Conference on Machine Learning and Applications, p. 7 (2005)
13. Hilario, M., Kalousis, A.: Fusion of meta-knowledge and meta-data for case-based model selection. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, pp. 180–191. Springer, Heidelberg (2001)
14. Kietz, J., Serban, F., Bernstein, A., Fischer, S.: Data mining workflow templates for intelligent discovery assistance and auto-experimentation. In: Third-Generation Data Mining: Towards Service-Oriented Knowledge Discovery SoKD 2010 (2010)
15. Kietz, J.-U., Serban, F., Bernstein, A., Fischer, S.: Towards cooperative planning of data mining workflows. In: Service-oriented Knowledge Discovery (SoKD 2009) Workshop at ECML/PKDD 2009 (2009)
16. Michie, D., Spiegelhalter, D., Taylor, C., Campbell, J.: Machine learning, neural and statistical classification (1994)
17. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale: Rapid prototyping for complex data mining tasks. In: KDD 2006: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 935–940. ACM, New York (2006)
18. Myers, K.: Strategic advice for hierarchical planners. In: Principles of Knowledge Representation and Reasoning-International Conference, pp. 112–123. Morgan Kaufmann Publishers, San Francisco (1996)
19. Nau, D.S.: May all your plans succeed (invited talk). In: Proceedings of the National Conference on Artificial Intelligence (AAAI) (July 2005)
20. Nau, D.S., Smith, S.J.J., Erol, K.: Control strategies in htn planning: Theory versus practice. In: IAAI Proceedings, pp. 1127–1133 (1998)
21. Serban, F., Kietz, J.-U., Bernstein, A.: An overview of intelligent data assistants for data analysis. In: Planning to Learn Workshop (PlanLearn 2010) at ECAI 2010, pp. 7–14 (2010)
22. Wirth, R., Shearer, C., Grimmer, U., Reinartz, T., Schlösser, J., Breitner, C., Engels, R., Lindner, G.: Towards process-oriented tool support for knowledge discovery in databases. In: Komorowski, J., Żytkow, J.M. (eds.) PKDD 1997. LNCS, vol. 1263, pp. 243–253. Springer, Heidelberg (1997)
23. Žáková, M., Křemen, P., Železný, F., Lavrač, N.: Planning to learn with a knowledge discovery ontology. In: Planning to Learn Workshop (PlanLearn 2008) at ICML 2008 (2008)
24. Žáková, M., Podpečan, V., Železný, F., Lavrač, N.: Advancing data mining workflow construction: A framework and cases using the orange toolkit. In: Service-oriented Knowledge Discovery (SoKD 2009) Workshop at ECML/PKDD 2009 (2009)