Semantic meta mining

Contents

1 From meta-learning to meta-mining

1.1 meta-learning

Meta-learning is learning to learn: in computer science, it is the application of machine learning techniques to meta-data describing past learning experience in order to modify some aspect of the learning process and improve the performance of the resulting model. Meta-learning thus defined applies specifically to learning, which is only one—albeit the central—step in the data mining (or knowledge discovery) process.

1.2 Rice model

The algorithm selection problem has its origins outside machine learning. In 1976 a seminal paper by John Rice proposed a formal model comprising four components (Fig. 1.): a problem space X or collection of problem instances describable in terms of features defined in feature space F, an algorithm space A or set of algorithms considered to address problems in X, and a performance space P representing metrics of algorithm efficacy in solving a problem. Algorithm selection can then be formulated as follows: Given a problem $x \in X$ characterized by $f(x) \in F$, find an algorithm $\alpha \in A$ via the selection mapping $S(f(x))$ such that the performance mapping $p(\alpha(x)) \in P$ is maximized.

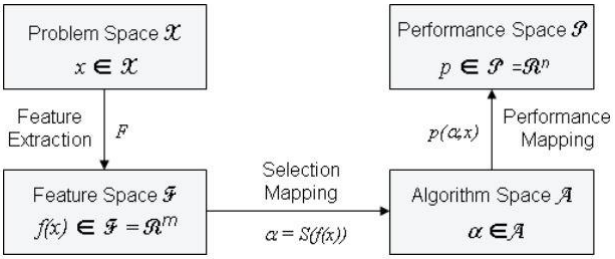

Fig. 1. Rice model

In Rice's model, selection mapping from problem space X onto algorithm space A is based solely on features $f \in F$ over the problem instances. In machine learning terms, the choice of the appropriate induction algorithm is conditioned solely on the characteristics of the learning problem and data. Strangely, meta-learning research has independently abided by the same restriction from its inception to the present. From early meta-learning attempts to more recent investigations, the dominant trend relies almost exclusively on meta-data describing the characteristics of base-level data sets used in learning, and the goal of meta-learning has even been defined restrictively as learning a mapping from dataset characteristics to algorithm performance. Research-

ers have come up with an abundant harvest of such characteristics, in particular statistical and information-theoretic properties of training data. A more recent research avenue, dubbed landmarking, characterizes data sets in terms of the predictive performance attained by simple learning algorithms when applied to them; yet another approach describes data sets based on features of the models that were learned from them. In all cases, however, the goal is to discover mappings from data set characteristics to learning algorithms viewed essentially as black boxes.

1.3 Revised Rice model

To overcome this difficulty, researchers proposed to extend the Rice framework and pry open the black box of algorithms. To be able to differentiate similar algorithms as well as to detect deeper commonalities among apparently unrelated ones, they propose to characterize them in terms of components such as the model structure built, the objective functions and search strategies used, or the type of data partitions produced. This compositional approach is expected to have two far-reaching consequences. Through a systematic analysis of all the ingredients that constitute an algorithm's inductive bias, meta-learning systems (and data miners in the first instance) will be able to infer not only which algorithms work for specific data approach to algorithm selection is not limited to the induction phase; it should be applicable to other data and model processing tasks that require search in the space of candidate algorithms. The proposed approach will also be adapted to model selection, i.e., finding the specific parameter setting that will allow a given algorithm to achieve acceptable performance on a given task. This will require an extensive study of the parameters involved in a given class of algorithms, their role in the learning process or their impact on the expected results (e.g., on the complexity of the learned model for induction algorithms), and their formalization in the data mining ontology.

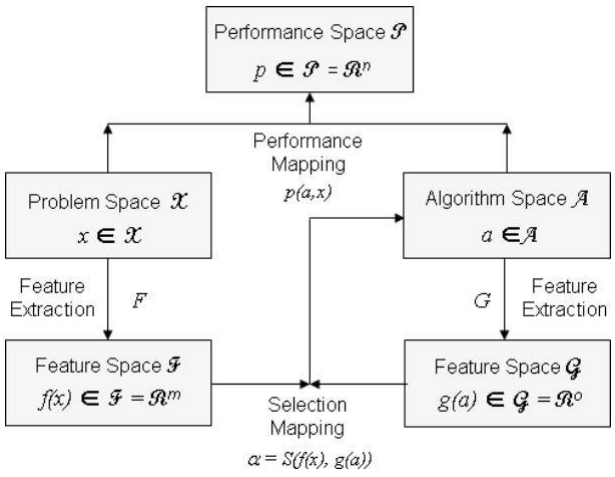

Fig. 2. The revised Rice's model

As Fig. 2. Shows, the proposed revision of Rice's model includes an additional feature space G representing the space of features extracted to characterize algorithms; selection mapping is now a function of both problem and algorithm features. The revised problem formulation now is: Given a problem $x \in X$ characterized by $f(x) \in F$ and algorithms $a \in A$ characterized by $g(a) \in G$, find an algorithm $\alpha \in A$ via the selection mapping $S(f(x), g(a))$ such that the performance mapping $p(a(x)) \in P$ is maximized.

1.4 Semantic meta mining

Semantic meta-mining is distinguished from traditional meta-learning by the following three properties. First, it extends the meta-learning approach to meta-mining, i.e. learning from the full DM process. Secondly, it is co-driven by knowledge of DM process and its components represented in the DM ontology and knowledge base (KB), in contrast to purely data driven traditional meta-learning. Thirdly, it breaks open the black box by explicitly analyzing DM algorithms along various dimensions to correlate observed performance of learned hypotheses resulting from DM processes with both data and algorithm characteristics. Semantic meta-mining is thus an ontology-based, process-oriented form of meta-learning that exploits in-depth knowledge of DM processes.

To support semantic meta-mining, the data mining ontologies contain a detailed taxonomy of algorithms used in DM processes, each described in terms of its underlying assumptions, the cost functions and adopted optimization strategies, the classes of hypotheses (models or pattern sets) it generates, and other properties. Following such a "glass box" approach makes explicit internal algorithm characteristics. This allows meta-learners using DM ontologies to generalize over algorithms and their properties, including those algorithms that did not appear in the training set.

2 Data Mining Ontologies

2.1 E-LICO

The goal of the e-LICO project is to build a virtual laboratory for interdisciplinary collaborative research in data mining and data-intensive sciences. The e-lab comprises three layers (Fig. 3): the e-science and data mining layers form a generic research environment that can be adapted to different scientific domains by customizing the application layer. e-LICO uses both Taverna and RapidAnalytics.

Fig. 3. The architecture of E-LICO

2. 2 Ontologies in E-LICO

The e-LICO data mining ontologies serve a number of different objectives. The first is to support planning of the knowledge discovery process and construction of workflows for a user task. This is pursued through the Data Mining Work Flow (DMWF) Ontology.

The second objective is to support algorithm, model and workflow selection for data mining tasks that require search in the space of possible methods and models. The third objective is to support meta-mining, or learning from past experience to improve data mining performance. The Data Mining Optimization (DMOP) Ontology has been developed in pursuit of these last two objectives.

2.3 OntoDM series

Panče Panov proposed a reference modular ontology for the domain of data mining OntoDM, directly motivated by the need for formalization of the data mining domain. The OntoDM ontology is designed and implemented by following ontology best practices and design principles. Its distinguishing feature is that it uses Basic Formal Ontology (BFO) as an upper-level ontology and a template, a set of formally defined relations from Relational Ontology (RO) and other state-of-the-art ontologies, and reuses classes and relations from the Ontology of Biomedical Investigations (OBI), the Information Artifact Ontology (IAO), and the Software Ontology (SWO). This will ensure compatibility and connections with other ontologies and allow cross-domain reasoning capabilities.

The OntoDM ontology is composed of three sub-ontologies covering different aspects of data mining:

Ontology of Datatypes (OntoDT), that supports the representation of knowledge about datatypes and is based on an accepted ISO standard for datatypes in computer systems;

Ontology of Core Data Mining Entities (OntoDM-core), that formalizes the key data mining entities for representing the mining of structured data in the context of a general framework for data mining; and

Ontology of Data Mining Investigations (OntoDM-KDD), that formalizes the knowledge discovery process based on the Cross Industry Standard Process for Data Mining (CRISP-DM) process model.

2.4 Other DM ontologies

There are several other data mining ontologies currently existing, such as the Knowledge Discovery (KD) Ontology, the OntoDTA ontology , the KDDONTO Ontology, the Data Mining Workflow (DMWF) Ontology , which are based on similar ideas.

3 An Ontology for Data Mining Optimization (DMOP)

The Data Mining OPtimization Ontology (DMOP) isdeveloped to support informed decision-making at various choice points of the data mining process. The ontology can be used as a reference by data miners and deployed in ontology-driven information systems. The primary purpose for which DMOP has been developed is the automation of algorithm and model selection through semantic meta-mining that makes use of an ontology-based meta-analysis of complete data mining processes in view of extracting patterns associated with mining performance. To this end, DMOP contains detailed descriptions of data mining tasks (e.g., learning, feature selection), data, algorithms, hypotheses such as mined models or patterns, and workflows. A development methodology was used for DMOP, including items such as competency questions and foundational ontology reuse. Several non-trivial modeling problems were encountered and due to the complexity of the data mining details, the ontology requires the use of the OWL 2 DL profile. DMOP was successfully evaluated for semantic meta-mining and used in constructing the Intelligent Discovery Assistant, deployed at the popular data mining environment RapidMiner.

3.1 DMOP: Architecture

In the e-LICO system, the information distilled in the DMOP ontology and knowledge base is used to drive the self-improvement of the DM process planner. Whereas the DMWF ontology supports the planner in its task of generating candidate workflows for a given mining task, the DMOP ontology supports the meta-miner, whose role is to analyse past DM experiments in order to build predictive models for ranking these workflows. The meta-miner is a semantic meta-miner: it draws its analytical power not only from metadata describing past experiments but also from back-

ground knowledge stored in the DMOP ontology and knowledge base. These meta-data are stored in RDF triple stores whose schemas are based on DMOP's conceptual framework. The figure shows the architecture of DMOP and its satellite databases that describe ingredients of DM experiments: operators (OPER-DB), datasets (DSET-DBs), workflows (WFLO-DB) and experimental parameters and results (DMEX-DBs). The architecture is presented in Fig. 4.

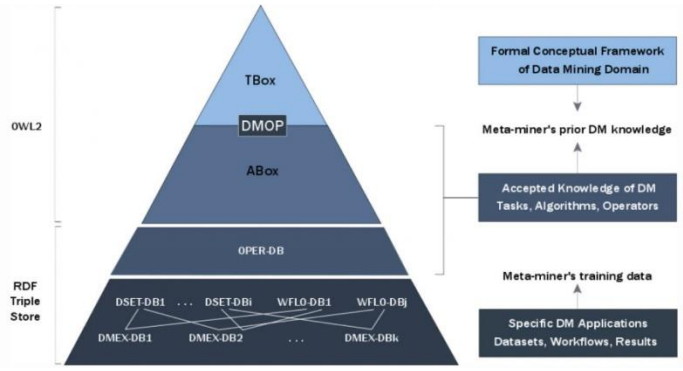

Fig. 4. DMOP Architecture

A distinctive feature of DMOP is its in-depth characterization of the major ingredients of the data mining process: datasets, algorithms, and learned hypotheses. Datasets are described using statistical and information-theoretic measures, as well as geometric complexity measures that suggest the potential difficulty of analysing them. Inductive paradigms and algorithms are modeled in terms of their implicit assumptions, their optimization strategies, their capabilities (e.g. ability to handle classification costs or instance weights), and their resilience to data flaws such as noise or missing values. Mined hypotheses are characterized by their structural complexity, interpretability, average performance in a given application domain, and -- for classifiers -- the type of decision boundaries induced in the instance space.

3.2 DMOP: content

The core concepts of DMOP are the different ingredients that go into the data mining process (DM-Process), which is shown in Fig. 5:

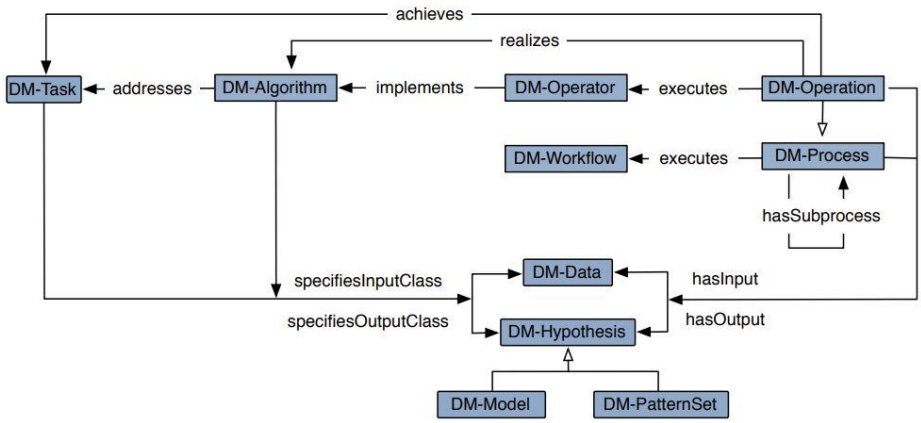

Fig. 5. DMOP content

- The input of the process is composed of a task specification (DM-Task) and training/test data (DM-Data) provided by the user;
- Its output is a hypothesis (DM-Hypothesis), which can take the form of a global model (DM-Model) or a set of local patterns (DM-PatternSet).
- Tasks and algorithms are not processes that directly manipulate data or models, rather they are specifications of such processes:
- A DM-Task specifies a DM process (or any part thereof) in terms of the input it requires and the output it is expected to produce.
- A DM-Algorithm is the specification of a procedure that addresses a given DM-Task, while a DM-Operator is a program that implements a given DM-Algorithm and that is executed by a DM-Operation.
- Instances of DM-Task and DM-Algorithm do no more than specifying their input/output types (only processes have actual inputs and outputs).

Some of the object properties of DM processes are:

- It hasInput and it hasOutput some IO-Object (DM-Data or DM-Hypothesis);
- A process that executes a DM operator also realizes the DM algorithm that isImplementedBy that operator;
- A DM algorithm addresses a DM task, and the process achieves the DM task addressed by the algorithm.

Finally, a DM-Workflow is a complex structure composed of DMoperators, and a DM-Experiment is a complex process composed of operations (or operator executions). An experiment is described by all the objects that participate in the process: a workflow, data sets used and produced by the different data processing phases, the resulting models, and meta-data quantifying their performance.

3.3 DMOP: DM-tasks

The top-level DM tasks listed below are defined by their inputs and outputs.

A DataProcessingTask receives and outputs data. Its four subclasses produce new data by cleansing (DataCleaningTask), reducing (DataReductionTask), extracting a compact representation (DataAbstractionTask) or otherwise transforming the input data (DataTransformationTask). These classes are further articulated in subclasses representing more fine-grained tasks.

An InductionTask consumes data and produces hypotheses. It can be either a ModelingTask or a PatternDiscoveryTask, based on whether it generates hypotheses in the form of global models or local pattern sets. Modeling tasks can be predictive (e.g. classification) or descriptive (e.g., clustering), while pattern discovery tasks are further subdivided into classes based on the nature of the extracted patterns: associations, dissociations, deviations, or subgroups.

A HypothesisProcessingTask consumes hypotheses and transforms (e.g., rewrites or prunes) them to produce enhanced—less complex or more readable—versions of the input hypotheses. A HypothesisEvaluationTask quantifies the quality of an induced.

3.4 DMOP: DM-Data

As the primary resource that feeds the knowledge discovery process, data have been a natural research focus for data miners. Over the past decades meta-learning researchers have actively investigated data characteristics that might explain generalization success or failure. Fig. 6 shows the characteristics associated with the different Data subclasses (shaded boxes). Most of these are statistical measures, such as the number of instances or the number of features of a data set. Others are information theoretic measures (italicized in the figure). Characteristics in bold font are geometric indicators of data set complexity, such as the maximum value of Fisher's Discriminant Ratio that measures the highest discriminatory power of any single feature in the data set.

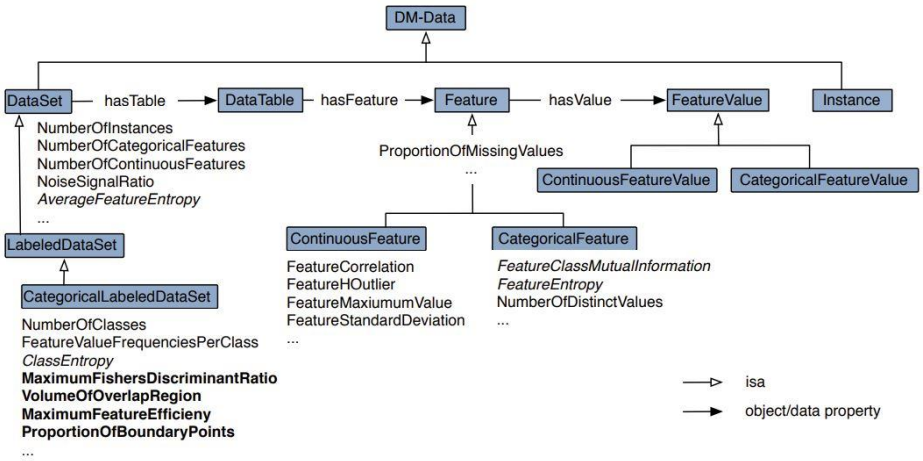

Fig. 6. DMOP: DM-Data

3.5 DMOP: DM-Algorithm

The top levels of the DM-Algorithm hierarchy reflect those of the DM-Task hierarchy, since each algorithm class is defined by the task it addresses (Fig. 7). However, the DM-Algorithm hierarchy is much deeper than the DM-Task hierarchy: for each leaf of the task hierarchy, there is often a dense subhierarchy of algorithms that specify diverse ways of addressing each task. For instance, the leaf concept ClassificationModelingTask maps directly onto the ClassificationModelingAlgorithm class.

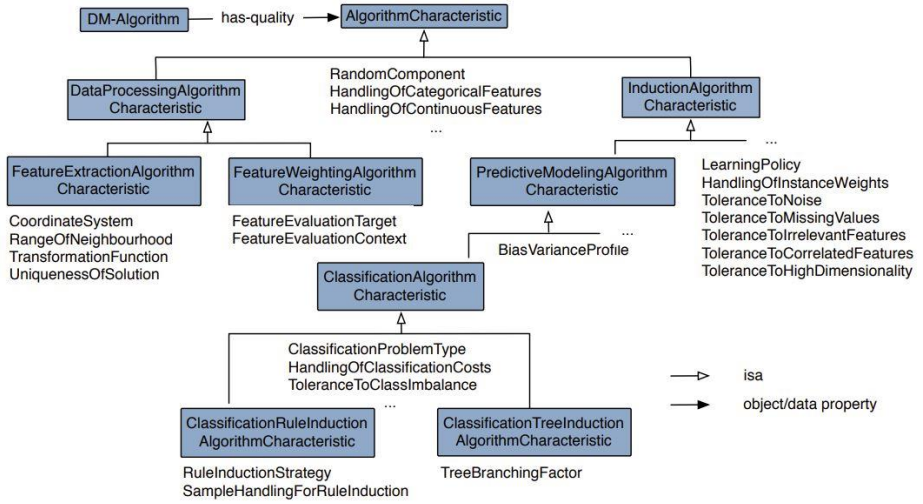

Fig. 7. DMOP: DM-Algorithm

A GenerativeAlgorithm computes the class-conditional densities p(Ck/x) and the priors p(Ck) for each class Ck. Examples of generative methods are normal (linear or quadratic) discriminant analysis and Naive Bayes.

A DiscriminativeAlgorithm, such as logistic regression, computes posterior probabilities p(Ck/x) directly to determine class membership.

A DiscriminantFunctionAlgorithm builds a direct mapping f(x) from input x onto a class label; neural networks and support vector classifiers (SVCs) are examples of discriminant function methods.

These three DM-Algorithm families spawn multiple levels of descendant classes that are distinguished by the type and structure of the models they generate. One innovative feature of DMOP is the modeling and exploitation of algorithm properties in meta-mining. All previous research in meta-learning has focused exclusively on data characteristics and treated algorithms as black boxes. DMOPbased meta-mining brings to bear in-depth knowledge of algorithms as expressed in their elaborate network of object properties. One of these is the object property has-quality, which relates a DM-Algorithm to an AlgorithmCharacteristic. A few characteristics are common to all DM algorithms; examples are characteristics that specify whether an algorithm makes use of a random component, or handles categorical or continuous features. Most other characteristics are subclass-specific. For instance, characteristics such as LearningPolicy (Eager/Lazy) are common to induction algorithms in general, whereas ToleranceToClassImbalance and andlingOfClassificationCosts make sense only for classification algorithms.

Note that has-quality is only one among the many object properties that are used to model DM algorithms. An induction algorithm, for instance, requires other properties to fully model its inductive bias. Some examples are the properties: assumes which expresses its underlying assumptions concerning the training data; specifiesOutputClass which links to the class of models generated by the algorithm, making explicit its hypothesis language or representational bias; hasOptimizationProblem which identifies its optimization problem and the strategies followed to solve it, thus defining its preference or search bias.

3.6 Usage of DMOP in semantic meta-mining

Today's DM platforms over many algorithm implementations (operators) that support different steps of the DM process. For instance, RapidMiner (version 5.3, Community Edition) offers 688 operators, either implemented by developers of RapidMiner or acquired through the implementation of wrappers for popular DM libraries such as Weka5. The user of the platform must select the appropriate operators, and their combination to build a DM workflow best addressing her goal. To assist the user in the design of an effective workflow, Intelligent Discovery Assistants (IDAs) have been proposed in Fig. 8.

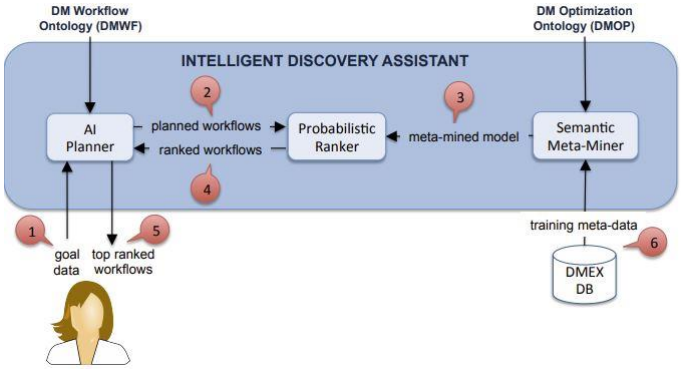

Fig. 8. Usage of DMOP in semantic meta-mining

The e-LICO IDA has the architecture of the so-called planning-based data analysis system since it uses artificial intelligence (AI) planning to construct a set of workflows. The planned workflows are all valid for the given task, but there may be many of them, even billions. Therefore, the planner based IDA exploits the results of semantic meta-mining to rank the workflows before they are presented to the user. The architecture of the IDA is shown in the figure. The user who interacts with the IDA is required to do no more than to upload annotated data (specifying roles and the types of the attributes) and to select the DM goal to be achieved (e.g. prediction) (1).
Data characteristics together with the DM Workflow Ontology (DMWF) are used by the IDA's AI-planner to generate a set of valid DM workflows (2).

Valid workflows are those that fulfill the user goal, take the dataset characteristic into account, and combine operators in the way that all their pre-conditions and post-conditions are met. Those workflows are passed to the probabilistic ranker that applies a default rule or a meta-mined model (3) computed by the semantic meta-miner to rank the workflows (4) which enables the AI planner to provide a list of top-ranked workflows to the user (5).

The workflows are ranked according to the estimated values of the performance measure of the DM hypotheses they produce (for instance, for a workflow addressing the classification task, accuracy can be such a measure). Best workflows, from the functional point of view, are those that achieve relatively best values for the measure. The meta-mined model is computed o↵-line by the metaminer, which is trained on a semantic repository of meta-data of data mining experiments (DMEX-DB) based on DMOP (6).