Semantic Meta-Mining

Part 3 of the
Tutorial on Semantic Data Mining

Melanie Hilario, Alexandros Kalousis
University of Geneva

Overview of Part 3

Melanie Hilario

- What is semantic meta-mining
- The meta-mining framework
- An ontology for semantic meta-mining
- A collaborative ontology development platform

Alexandros Kalousis

- From meta-learning to semantic meta-mining
- Semantic meta-mining
- Semantic meta-mining for DM workflow planning

Appendix: Selected bibliography

What is meta-learning

- Learning to learn: use machine learning methods to improve learning

	Base-level learning	Meta-level learning
Application domain	any	machine learning
Ex. learning tasks	diagnose disease, predict stocks prices	select learning algorithm, parameters
Training data	domain-specific observations	meta-data from learning experiments

- Dates back to the 1990's (see Vilalta, 2002 for a survey)
- Strong tradition in Europe via successive EU projects: StatLog, Metal, e-LICO

Limitations of traditional meta-learning

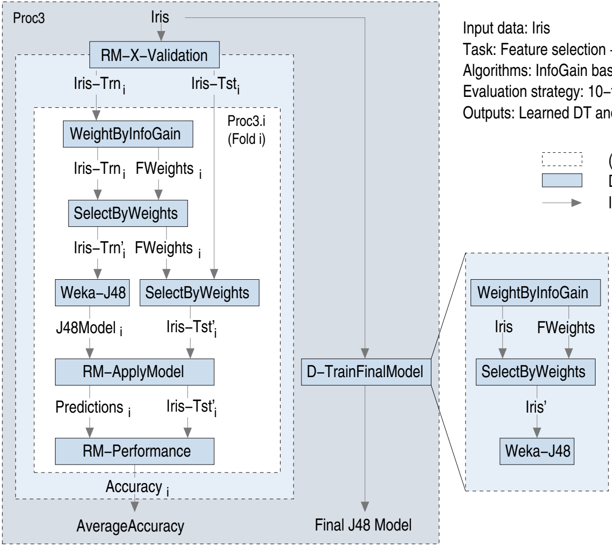
Our focus: data mining (DM) optimization via algorithm/model selection

- Implicitly bound to the **Rice model** for algorithm selection
  - $\Rightarrow$ Based solely on data characteristics.
  - $\Rightarrow$ Algorithms treated as black boxes.
- **Greedy**: Restricted to the current (usually inductive) step of the DM process
- **Purely data-driven**: No integration of explicit DM knowledge into meta-learning

Beyond meta-learning

- **Revised Rice model**: break the algorithmic black box
  Use both dataset and algorithm characteristics to meta-learn
- **Meta-mining**: process-oriented meta-learning
  Rank/select workflows rather than individual algorithms/parameters
- **Semantic meta-mining**: ontology-driven meta-mining
  Incorporate specialized knowledge of algorithms, data and workflows from a DM ontology

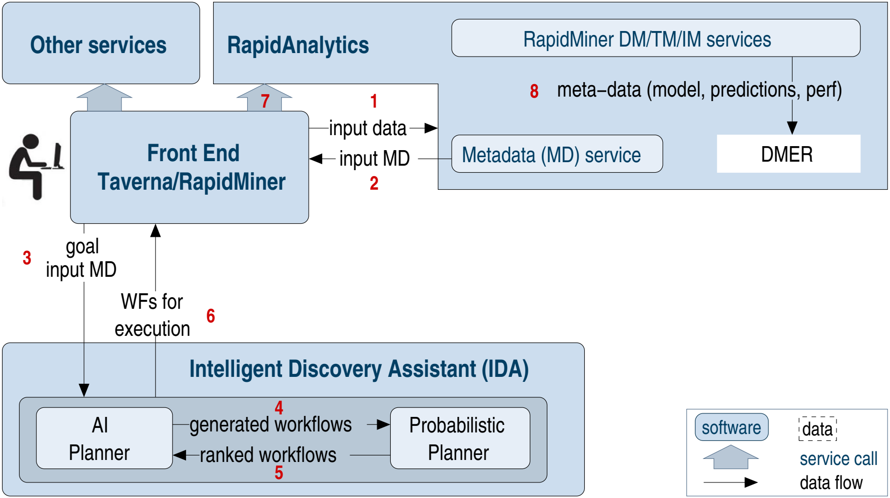
Example of a DM Workflow



Input data: Iris
Task: Feature selection + classification
Algorithms: InfoGain based FS + DT
Evaluation strategy: 10–fold cross–val
Outputs: Learned DT and estimated accuracy

(Sub–)Workflows
DM Operators (nodes)
Inputs/outputs (edges)

The data mining context



The data-mining context (comments)

The user inputs a DM goal and an input dataset from either the Taverna or the RapidMiner front end.

1-2. RapidAnalytics' MD service extracts meta-data to be used by the AI Planner.

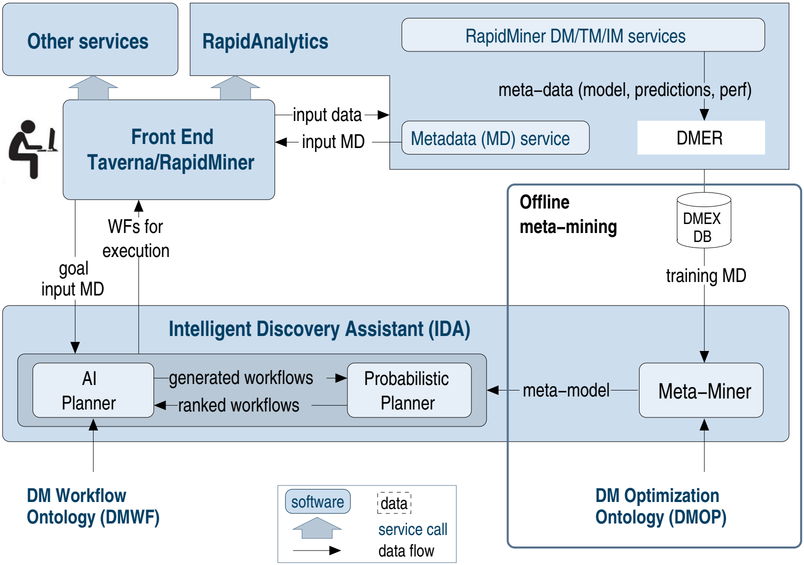
3-4. The IDA's basic AI Planner generates applicable workflows in a brute force fashion.

5. The Probabilistic Planner ranks the workflows **based on lessons drawn from past DM experience.**

6-7. The selected WFs are sent to RapidMiner for execution.

8. All process predictions, models, and meta-data are stored in the Data Mining Experiments Repository (DMER)

How the IDA becomes intelligent



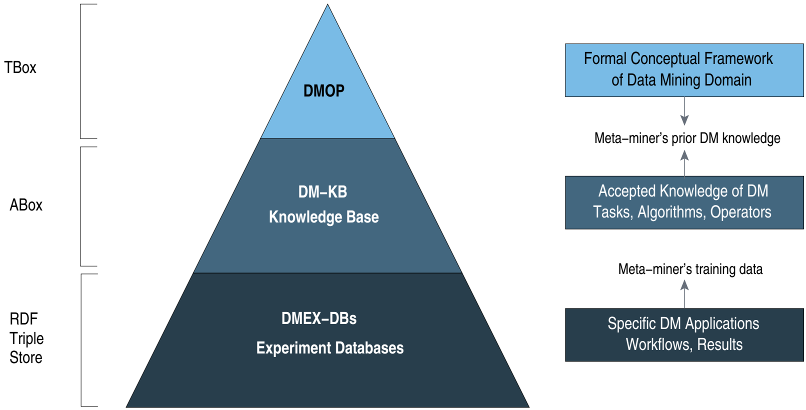
How the IDA becomes intelligent (comments)

- Selected meta-data from the DM Experiment Repository are structured and stored in the DMEX-DB
- Training data in DMEX-DB represented using concepts from the DM Optimization Ontology (DMOP)
- The meta-miner extracts workflow patterns and builds predictive models using
  - training data from DMEX-DB
  - prior DM knowledge from DMOP

DMOP: Data Mining OPtimization ontology

- **Purpose**: structure the space of DM tasks, data, models, algorithms, operators and workflows
  - ⇒ higher-order feature space in which meta-learning can take place
- **Approach**: model algorithms in terms of their underlying assumptions and other components of bias
  - ⇒ allows for generalization over algorithms and hence over workflows
  - ⇒ supports **semantic meta-mining**

Structure of DMOP



Structure of DMOP (comments)

DMOP (TBox):

- a comprehensive conceptual framework for describing data mining objects and processes (p. 14)
- detailed sub-ontologies of classification, pattern discovery and feature extraction/weighting/selection algorithms

  ⇒ illustrate our approach to breaking the algorithmic black box (p. 15)

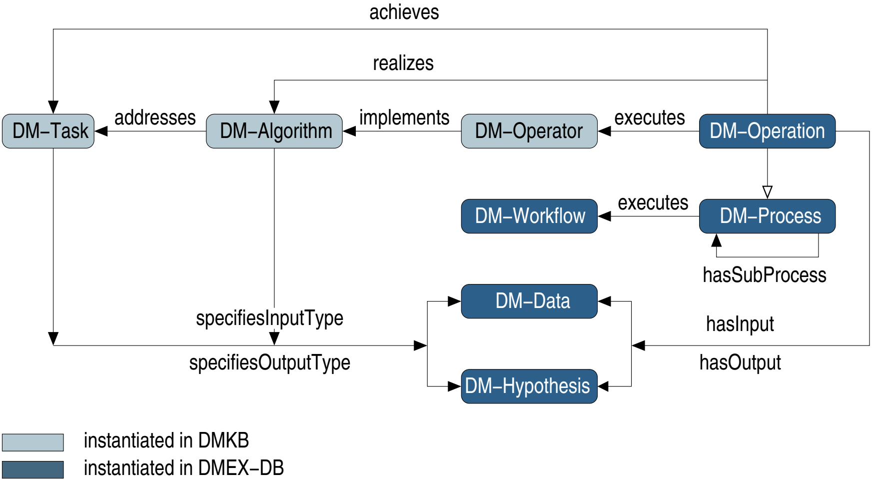
  ⇒ will serve as models for annotating new DM algorithm families

DM-KB (ABox)

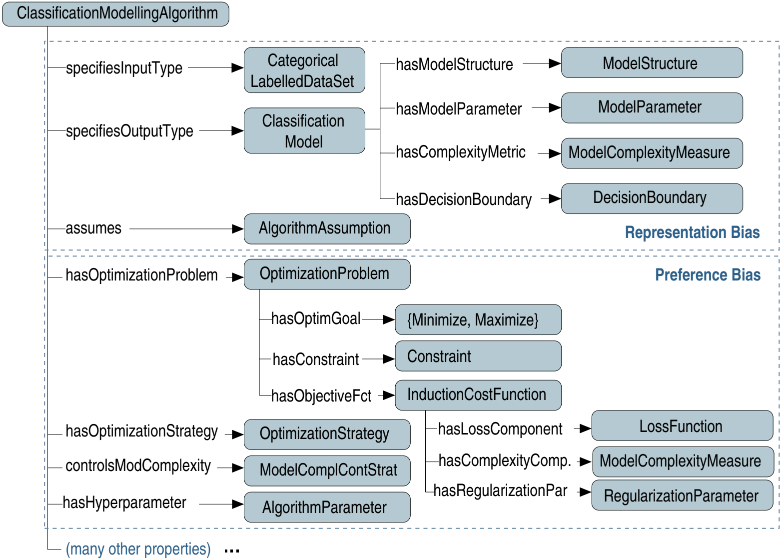
- describes individual algorithms using concepts from DMOP
- links available operators from known DM packages to their source algorithms

  ⇒ generalized frequent pattern mining over WFs from DMER

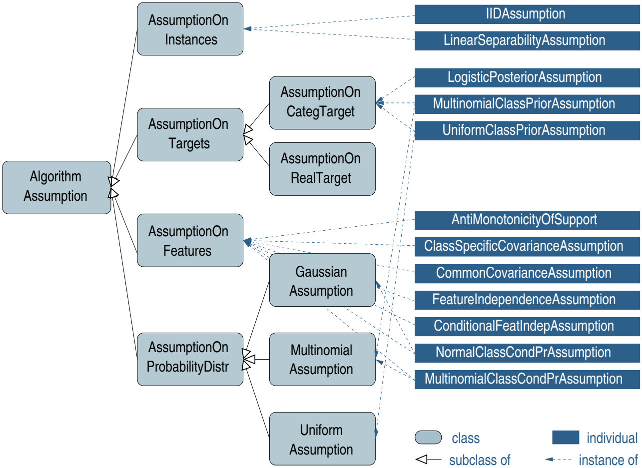
The Conceptual Framework



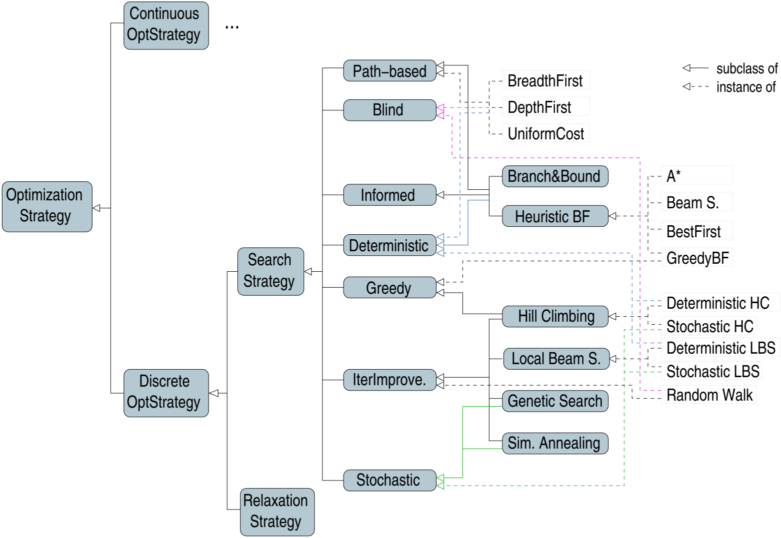
Inside Induction Algorithms



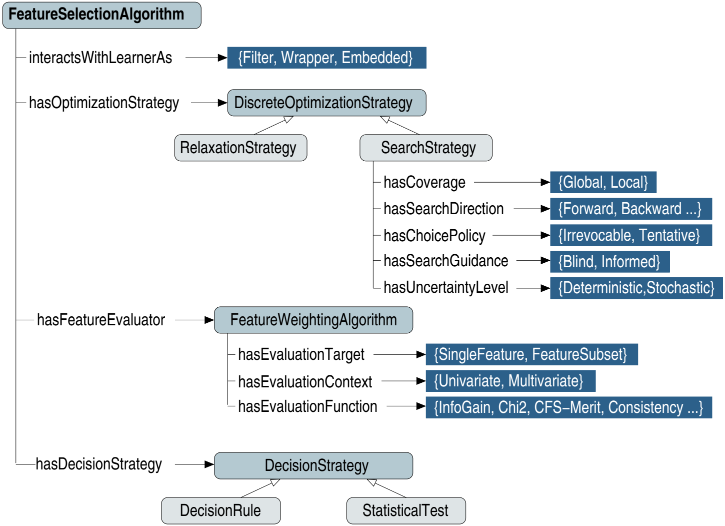
Algorithm Assumptions



Optimization Strategies



Feature Selection and Weighting

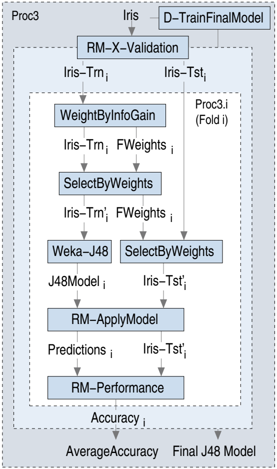


Example: Correlation-Based Feature Selection

CorrelationBasedFeatureSelection

- hasOptimizationStrategy → **GreedyForwardSelection**
  - hasCoverage → **Global**
  - hasSearchDirection → **Forward**
  - hasChoicePolicy → **Irrevocable**
  - hasSearchGuidance → **Informed**
  - hasUncertaintyLevel → **Deterministic**
- hasFeatureEvaluator → **CFS–FWA**
  - hasEvaluationTarget → **FeatureSubset**
  - hasEvaluationContext → **Multivariate**
  - hasEvaluationFunction → **CFS–Merit**
- hasDecisionStrategy → **CFS–SearchStopRule**
  - hasDecisionTarget → **FeatureSubsetWeight**
  - hasDecisionCriterion → **NumCyclesNoImprovement**
  - hasRelationalOp → **EqRelOp**
  - hasFixedThreshold → **5**

Modeling Workflows in DMOP




```
Proc3: DM-Process
hasInput(Proc3, Iris)
executes(Proc3, FSC-Infogain-J48-Xval-Wf)
hasOutput(Proc3, J48Model3-Final)
hasOutput(Proc3, AvgAccuracy)
hasFirstSubprocess(Proc3, Opex3-Xval)
hasSubProcess(Proc3, Opex3-Xval)
hasSubProcess(Proc3, Opex3-TrainFinalModel)

    Opex3-Xval: DM-Operation
    hasFirstSubprocess(Opex3-Xval, Proc3.i)
    executes(Opex3-Xval, RM-X-Validation)
    hasParameterSetting(Opex3-Xval, OpSet3)
    hasOutput(Opex3-Xval, AvgPerfMeasure3)
    isFollowedDirectlyBy.{OpEx3-TrainFinalModel)
    isFollowedBy(OpEx3-TrainFinalModel)
    isSubprocessOf(Opex3-Xval, Proc3)
    hasSubProcess(Opex3-Xval, Proc3.i)

        Proc3.i: DM-Process
        hasInput(Proc3.i, Iris-Trn3.i)
        hasInput(Proc3.i, Iris-Tst3.i)
        hasOuptut(Proc3.i, PerfMeasure-3.1.fold-i)
        hasFirstSubprocess(Proc3.i, Opex3.i.1-WeightByInfogain)
        isSubprocessOf(Proc3.i, Opex3-Xval)
        hasSubProcess(Proc3.i, Opex3.i.1-WeightByInfogain)
        hasSubProcess(Proc3.i, Opex3.i.2-SelectByWeights)
        hasSubProcess(Proc3.i, Opex3.i.3-J48)
        hasSubProcess(Proc3.i, Opex-3.i.4-SelectByWeights)
        hasSubProcess(Proc3.i, Opex3.i.5-ApplyModel)
        hasSubProcess(Proc3.i, Opex3.i.6-Performance)
        ...
```

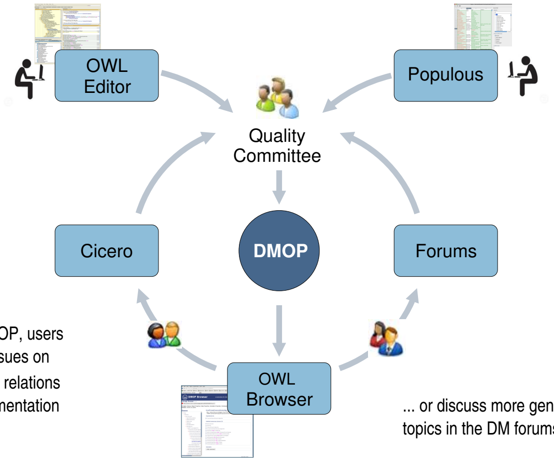
The DMOP CODeP

Mode 1

Ontology–savvy DM experts develop DMOP sub–ontologies directly on OWL editors

Mode 2

The Populous tool allows data miners to help populate DMOP by filling pre–defined templates.



Mode 3

While browsing DMOP, users raise and resolve issues on specific concepts or relations via the Cicero argumentation platform ...

... or discuss more general topics in the DM forums

Towards a DMO Foundry

- There is a growing body of data mining ontologies: KD Ontology, DMWF, OntoDM, KDDOnto, Exposé.
- The goal of the DMO Foundry is to serve as a portal for exploration and collaborative development of these ontologies.
- Each participating ontology will have its own CODeP.
- DMOP currently used to seed the DMO Foundry: all volunteers welcome!
- Visit http://www.dmo-foundry.org and register for a login.

How DMOP supports meta-mining

- provides a unified framework for describing DM processes, data, algorithms, and mined hypotheses (models and pattern sets)
- breaks open the black box of algorithms and analyses their components, capabilities and assumptions
- provides prior DM knowledge that allows the meta-miner to extract meaningful workflow patterns and correlate them with expected performance.

  $\Rightarrow$ How this is done is described in the next talk of this tutorial.

Overview of Part 3

Melanie Hilario

- What is semantic meta-mining
- The meta-mining framework
- An ontology for semantic meta-mining
- A collaborative ontology development platform

Alexandros Kalousis

- From meta-learning to semantic meta-mining
- Semantic meta-mining
- Semantic meta-mining for DM workflow planning

Appendix: Selected bibliography

Standard meta-learning

- The typical meta-learning problem formulation would construct performance predictive models:
  - for a specific algorithm
  - for specific couples of algorithms
  - for specific sets of algorithms
- given some collection of datasets to which these algorithms were applied
- relying only on DCs and the algorithms performance measures
- A typical meta-learning model can only make predictions for the *specific* algorithms on which it was trained.

Moving ahead from meta-learning

- *Standard meta-learning* typically relies on the use of Dataset Characteristics, DC, only

$\Downarrow$ DMOP ontology

- we can now do *sematic meta-learning* where in addition to DC we also have algorithm and Data Mining Algorithm and Operator characteristics given by the DMOP.

Semantic meta-learning

- A semantic meta-learning problem would associate *Algorithms Descriptors* with *Dataset Characteristics* based on *performance measures*
- given some collection of datasets to which some algorithms were applied
- relying on DCs, the Algorithms Descriptors, and the algorithms performance measures
- A semantic meta-learning model can in principle make performance predictions for algorithms other than the ones on which it was created as long as the former are described in the DMOP.
- Very similar in nature to collaborative/content based filtering problems

Semantic meta-learning: a first effort

- We did some very preliminary steps in [2] using semantic kernels to exploit the semantic descriptors of the algorithms provided by the DMOP.
- These kernels where combined with a similarity measure on dataset characteristics and derived a final similarity measure, defined over pairs of the form *(algo, dataset)*.
- The similarity measure was used in a nearest neighbor algorithm to predict whether a specific match was good (high expected predictive performance) or not.
- The incorporation of algorithms semantic descriptors seemed to improve the predictive performance.

Semantic meta-mining

- Semantic meta-mining differs from its meta-learning counterpart in that we are acting on *workflows* of data mining operators/algorithms.

Semantic Meta-mining

We will present the following use cases of semantic meta-mining

- mining for frequent generalized patterns over workflow collections to be used for:
  - workflow description
  - worflow planning
- looking for associations between *DM workflow characteristics* and *dataset characteristics* based on *performance measures*.

In all of them the use of the DMOP is central

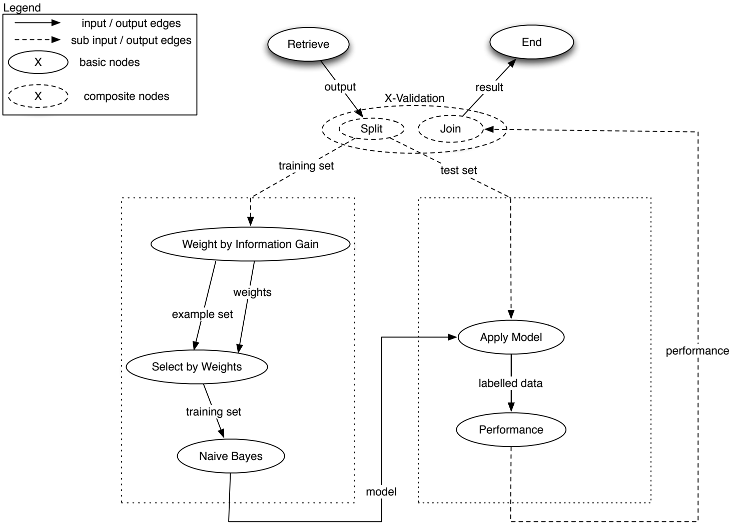
Data mining workflows representation

- DM wfs are Hierarchical Directed Acyclic Graphs in which:
  - nodes are Data Mining operators representing the control flow
  - edges are Input/Output objects representing the data flow
- We want to be able to mine *generalized* workflow patterns, i.e. patterns that do not contain only ground operators but also abstract classes of operators, exploiting the hierarchies of the DMOP.
- working with the *parse tree* representation of the DM workflows, representing the topological sort of the HDAG, is a natural choise.

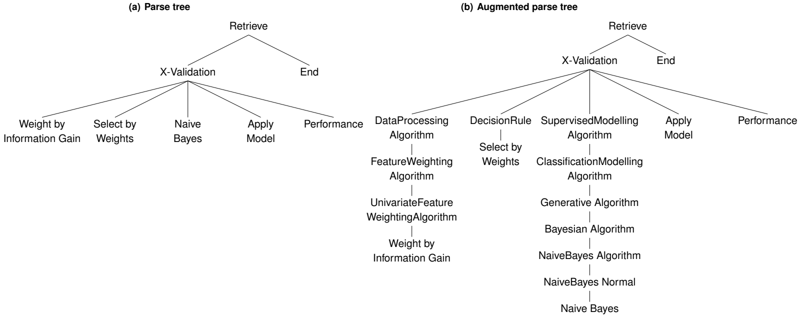
Frequent generalized pattern mining over workflows I

- From a data mining workflow derive
- a parse tree and from that derive
- an augmented parse tree by including these parts of the DMOP that describe the operators of the WF
- pattern mining will take place over the augmented parse tree representations
- the resulting patterns produce a new propositional representation of the workflows that includes the DMOP information

A Data Mining Workflow



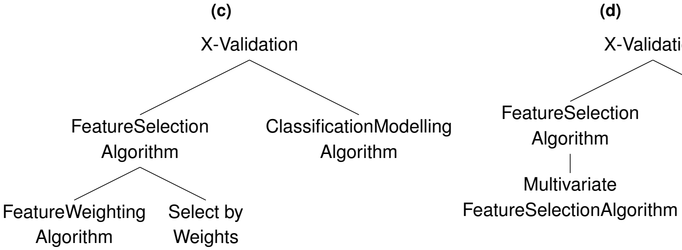
Parse and augmented parse tree of the previous WF



Generalized Frequent Pattern Extraction Results

- 28 data mining workflows, combinations of feature selection (four) with classification algorithms (seven).
- 456 augmented trees.
- Using TreeMiner, [1], with a support of 3% we got 1052 generalized closed patterns.
- Each of the 28 workflows can now be described by the presence/absence of the 1052 patterns in it.

Some Examples of Generalized Workflow Patterns



Meta-mining: associating workflow with dataset characteristics for performance prediction

The setting:

- 28 data mining workflows, applied on
- 65 cancer microarray classification problems with
- performance estimates acquired by 10-fold cross-validation.
- A total of 1820 base-level data mining experiments.
- Each *experiment=(wf, dataset)* was assigned a label from {*best, rest*} based on a statistical significance test (class distribution: 45% best, 55% rest).

The goal:

- find combinations of workflow and dataset characteristics that are associated with high predictive performance (*best* label).

Meta-mining: associating workflow with dataset characteristics for performance prediction (contd.)

- Workflows are described by the presence/absence of the 1052 closed patterns
- Datasets are described by a set of 18 statistical, information-based, and geometrical features.
- We learn a model by simply applying a decision tree algorithm on the DM experiments description.
- Different evaluation scenarios:
  - leave-one-dataset out
  - leave-one-dataset-workflow out (to see whether we can make predictions on the performance of workflows that were never seen)
- In both scenarios we get a performance improvement over the baseline of default accuracy

Meta-mining for DM workflow planning

- Equip a basic AI planner that follows the CRISP-DM model with a meta-mined model that will guide task/method/operator selection in view of optimizing some performance measure

Basic challenge

Given:

- a dataset $\mathbf{d}$
- a data mining goal g
- a set of data mining operators $\mathbf{O}$
- some target performance measure a that we want to optimize

plan a data mining workflow,

$$WF = [S_1, S_2, \ldots S_n], S_i \in \mathbf{O}$$

that will have the maximum probability of been observed, i.e.

$$\begin{aligned} WF &:= arg \max_{WF} P(S_1, S_2, \ldots S_n | \mathbf{d}, g, a) \\ &= arg \max_{WF} P(S_1 | \mathbf{d}, g, a) \prod_{i=2}^{N} P(S_i | S_{i-1}, \mathbf{d}, g, a) \end{aligned}$$

The AI-planner

- Is a Hierarchical Task Network decomposition planner
- which creates hierarchical, tree-like, plans using task and method decompostions.
- At each expansion point it needs support on which task or method or operator it should select given:
  - the so far constructed sequence of operators $W_{i-1} = [o_1, o_2, \ldots, o_{i-1}]$
  - the tasks and methods that these operators achieve given by the so far constructed HTN tree Tr_{i-1}
  - the current state S_{i-1}, namely the set of available I/O objects
  - the g planning goal
- this support is provided by a meta-mined state transition matrix.

State transition matrix

- The planner relies on a meta-mined state transition matrix $\mathbf{T}$ with size: $|\mathbf{O}| \times |\mathbf{O}|$, where
$$T_{ij} = P(o_j | o_i, \mathbf{d}, g, a)$$
- this will be learned from past experiences and we will do so with meta-mining

Modelling the transition matrix

- Original idea focus on transitions of the form $P(o_i|o_j)$.
- However such short transitions are not appropriate for DM workflows so instead we will use the transition probability:

$$P(o_i = o|W_{i-1}, S_{i-1}, Tr_{i-1}, g)$$

- which is equivalent to computing the confidence of the association rule:

$$W_{i-1} \rightarrow o$$

which is given by:

$$\frac{support(W_i^o = W_{i-1} \cup \{o\})}{support(W_{i-1})} = P(o_i = o|W_{i-1})$$

W_i^o is the workflow that we get if we add operator o to W_{i-1}

Selecting which o operator to apply

- Given a so far workflow W_{i-1} we need to compute

$$arg \max_{o} P(o_i = o | W_{i-1}, S_{i-1}, Tr_{i-1}, g)$$

- this requires *exact* matching of W_{i-1} against the collection of previously applied workflows, *overly specific* and most probably will return a no-match.
- We relax this matching and use instead a partial one using frequent workflow patterns.
- Let $C = \{fp_i | support(fp_i) \geq \theta\}$ a collection of frequent workflow patterns extracted from some data mining workflow collection.

Selecting which o operator to apply using frequent patterns

- Look for frequent patterns $fp \in C$ such that:

$$fp \in W_i^o \text{ and } o \in fp$$

- and compute:

$$p(o_i = o | fp - \{o\}) = \frac{support(fp)}{support(fp - \{o\})}$$

- use the quality measure:

$$q(o) = (p(o_i = o | fp - \{o\}) + \lambda \times support(fp - \{o\}))$$

  trading off confidence for support, according to λ

- and select the o operator according to:

$$arg \max_o q(o)$$

Accounting for the workflows' performance measures

- We adapt the above idea to account for performance, e.g. predictive accuracy
  - Base-level mining experiments are divided in two classess, namely high predictive performance, H, and low predictive performance, L
  - Select operators according to:

$$arg \max_{o} \frac{q_H(o)}{q_L(o)}$$

i.e. with maximal quality in the high performance class and minimal in the low.

Accounting for the dataset characteristics

A number of solutions:

- Cluster the space of datasets to performance aware clusters using the dataset characteristics
  - Situate a dataset in its respective cluster and then use the cluster specific $\frac{q_H(o)}{q_L(o)}$ estimates
- Modify the computation of support to reflect dataset similarities and not just counts
  - Drawback: requires recomputation of the frequent patterns each time a new dataset appears.

Current Status

- Operational system
- Evaluating the different approaches
- Many different future directions, especially on how one can use the rich information provided by DMOP to meta-mine.

Bibliography I

On semantic meta-mining

[1] M. Hilario, P. Nguyen, H. Do, A. Woznica, and A. Kalousis. Ontology-based meta-mining of knowledge discovery workflows. In K. Grabczewski N. Jankowski, W. Duchs, editor, *Meta-Learning in Computational Intelligence*, pages 273–316. Springer, 2011.

[2] D. T. Wijaya, A. Kalousis, and M. Hilario. Predicting Classifier Performance using Data Set Descriptors and Data Mining Ontology. In Proceedings of the Planning to learn Workshop, ECAI-2010.

On data mining ontologies

[1] M. Cannataro and C. Comito. A data mining ontology for grid programming. In *Proc. 1st Int. Workshop on Semantics in Peer-to-Peer and Grid Computing, in conjunction with WWW-2003*, pages 113–134, 2003.

[2] C. Diamantini, D. Potena, and E. Storti. Supporting users in KDD process design: A semantic similarity matching approach. In *Proc. 3rd Planning to Learn Workshop (in conjunction with ECAI-2010)*, pages 27–34, Lisbon, 2010.

[3] M. Hilario, A. Kalousis, P. Nguyen, and A. Woznica. A data mining ontology for algorithm selection and meta-learning. In *Proc. ECML/PKDD Workshop on Third-Generation Data Mining: Towards Service-Oriented Knowledge Discovery (SoKD-09)*, Bled, Slovenia, September 2009.

[4] J.-U. Kietz, F. Serban, A. Bernstein, and S. Fischer. Data mining workflow templates for intelligent discovery assistance and auto-experimentation. In *Proc. 3rd Workshop on Third-Generation Data Mining: Towards Service-Oriented Knowledge Discovery (SoKD-10)*, pages 1–12, 2010.

[5] P. Panov, L. Soldatova, and S. Dzeroski. Towards an ontology of data mining investigations. In *Discovery Science*, 2009.

[6] Joaquin Vanschoren and Larisa Soldatova. Exposé: An ontology for data mining experiments. In *International Workshop on Third Generation Data Mining: Towards Service-oriented Knowledge Discovery (SoKD-2010)*, September 2010.

[7] M. Zakova, P. Kremen, F. Zelezny, and N. Lavrac. Automating knowledge discovery workflow composition through ontology-based planning. *IEEE Transactions on Automation Science and Engineering*, 2010.

Bibliography II

On meta-learning

[1] M. L. Anderson and T. Oates. A review of recent research in metareasoning and metalearning. *AI Magazine*, 28(1):7–16, 2007.

[2] H. Bensusan and C. Giraud-Carrier. Discovering task neighbourhoods through landmark learning performances. In *Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 325–330, 2000.

[3] P. Brazdil, J. Gama, and B. Henery. Characterizing the applicability of classification algorithms using meta-level learning. In *Machine Learning: ECML-94. European Conference on Machine Learning*, pages 83–102, Catania, Italy, 1994. Springer-Verlag.

[4] W. Duch and K. Grudzinski. Meta-learning: Searching in the model space. In *Proc. of the Int. Conf. on Neural Information Processing (ICONIP), Shanghai 2001*, pages 235–240, 2001.

[5] J. Fürnkranz and J. Petrak. An evaluation of landmarking variants. In *Proceedings of the ECML Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-learning*, pages 57–68, 2001.

[6] C. Giraud-Carrier, R. Vilalta, and P. Brazdil. Introduction to the special issue on meta-learning. *Machine Learning*, 54:187–193, 2004.

[7] A. Kalousis. *Algorithm Selection via Meta-Learning*. PhD thesis, University of Geneva, 2002.

[8] B. Pfahringer, H. Bensusan, and C. Giraud-Carrier. Meta-learning by landmarking various learning algorithms. In *Proc. Seventeenth International Conference on Machine Learning, ICML'2000*, pages 743–750, San Francisco, California, June 2000. Morgan Kaufmann.

[9] K. A. Smith-Miles. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys*, 41(1), 2008.

[10] C. Soares and P. Brazdil. Zoomed ranking: selection of classification algorithms based on relevant performance information. In *Principles of Data Mining and Knowledge Discovery. Proceedings of the 4th European Conference (PKDD-00*, pages 126–135. Springer, 2000.

[11] C. Soares, P. Brazdil, and P. Kuba. A meta-learning method to select the kernel width in support vector regression. *Machine Learning*, 54(3):195–209, 2004.

[12] R. Vilalta and Y. Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18:77–95, 2002.

[13] R. Vilalta, C. Giraud-Carrier, P. Brazdil, and C. Soares. Using meta-learning to support data mining. *International Journal of Computer Science and Applications*, 1(1):31–45, 2004.

Bibliography III

Other

[1] M.J. Zaki Efficiently mining frequent trees in a forest: Algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering*, 17:1021–1035, special issue on Mining Biological Data.