Semantic Data Mining

Tutorial at ECML/PKDD 2011

Athens

September 9, 2011

Tutorial overview

- Part 1: Introduction to Semantic Data Mining (SDM)
  Nada Lavrac, Anze Vavpetic
  Jozef Stefan Institute, Ljubljana, Slovenia

- Part 2: Learning from Description Logics (DL-learning)
  Agnieszka Lawrynowicz, Jedrzej Potoniec
  Poznan University of Technology, Poznan, Poland

- Part 3: Semantic meta-mining
  Melanie Hilario, Alexandros Kalousis
  University of Geneva, Geneva, Switzerland

Introduction to Semantic Data Mining (SDM)

Nada Lavrac

- Background and motivation
- What is Semantic Data Mining: Definition and settings
- Early work in Semantic subgroup discovery

Anze Vavpetic

- …

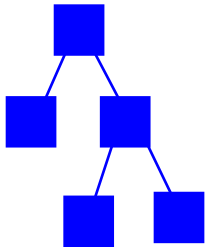
Background and motivation: Data mining

Person	Age	Spect. presc.	Astigm.	Tear prod.	Lenses
O1	young	myope	no	reduced	NONE
O2	young	myope	no	normal	SOFT
O3	young	myope	yes	reduced	NONE
O4	young	myope	yes	normal	HARD
O5	young	hypermetrope	no	reduced	NONE
O6-O13	...	...	...	...	...
O14	pre-presbyo	hypermetrope	no	normal	SOFT
O15	pre-presbyo	hypermetrope	yes	reduced	NONE
O16	pre-presbyo	hypermetrope	yes	normal	NONE
O17	presbyopic	myope	no	reduced	NONE
O18	presbyopic	myope	no	normal	NONE
O19-O23	...	...	...	...	...
O24	presbyopic	hypermetrope	yes	normal	NONE

data

knowledge discovery from data

Data Mining



model, patterns, …

Given: transaction data table, a set of text documents, …

Find: a classification model, a set of interesting patterns

Background and motivation: Using BK in data mining

- Using background knowledge in data mining has been a topic of extensive research
  - Hierarchical attribute values (Michalski et al. 1986,…), hierarchy/taxonomy of attributes, …
  - ILP (Muggleton, 1991; Lavrac and Dzeroski 1994), relational learning (Quinlan, 1993), propositionalization (Lavrac et al. 1993), …

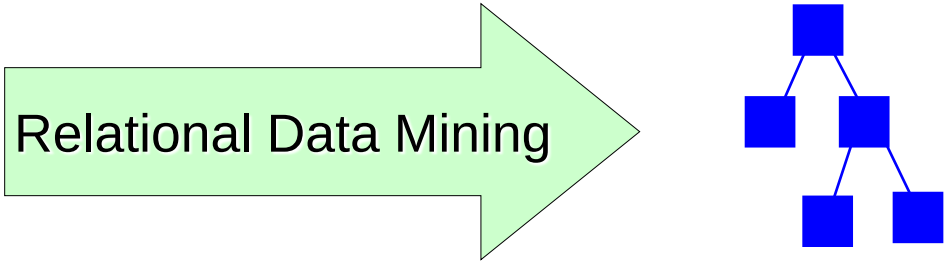
Background and motivation: Relational data mining

customer							
ID	Zip	Sex	So St	In come	Age	Club	Resp
...	...	...	...	...	...	...	...
3478	34677	m	si	60-70	32	me	nr
3479	43666	f	ma	80-90	45	nm	re
...	...	...	...	...	...	...	...

order				
Customer ID	Order ID	Store ID	Delivery Mode	Paymt Mode
...	...	...	...	...
3478	2140267	12	regular	cash
3478	3446778	12	express	check
3478	4728386	17	regular	check
3479	3233444	17	express	credit
3479	3475886	12	regular	credit
...	...	...	...	...

store			
Store ID	Size	Type	Location
...	...	...	...
12	small	franchise	city
17	large	indep	rural
...	...	...	...

Relational representation of customers, orders and stores.



model, patterns, …

Given: a relational database, a set of tables, sets of logical facts, a graph, …

Find: a classification model, a set of patterns

Background and motivation: Relational data mining

- **ILP, relational learning, propositionalization**
  - Learning from complex multi-relational data

customer

ID	Zip	Sex	So St	In come	Age	Club	Resp
...	...	...	...	...	...	...	...
3478	34677	m	si	60-70	32	me	nr
3479	43666	f	ma	80-90	45	nm	re
...	...	...	...	...	...	...	...

order

Customer ID	Order ID	Store ID	Delivery Mode	Paymt Mode
...	...	...	...	...
3478	2140267	12	regular	cash
3478	3446778	12	express	check
3478	4728386	17	regular	check
3479	3233444	17	express	credit
3479	3475886	12	regular	credit
...	...	...	...	...

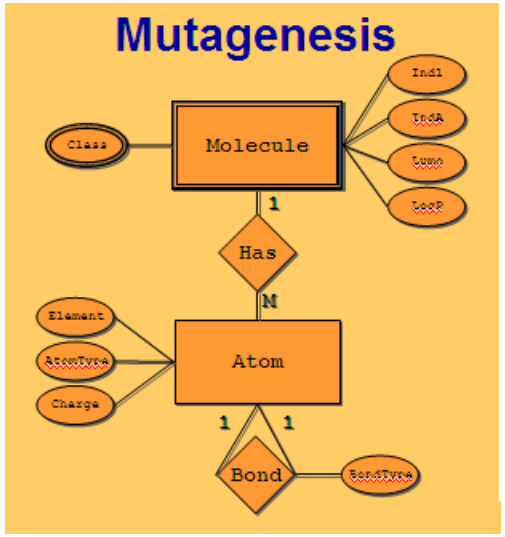
store

Store ID	Size	Type	Location
...	...	...	...
12	small	franchise	city
17	large	indep	rural
...	...	...	...

Relational representation of customers, orders and stores.

Background and motivation: Relational data mining

- **ILP, relational learning, propositionalization**
  - Learning from complex multi-relational data
  - Learning from complex structured data: e.g., molecules and their properties in protein engineering, biochemistry, ...



customer

ID	Zip	Sex	Soc St	Income	Age	Club	Resp
...	...	...	...	...	...	...	...
3478	34677	m	si	60-70	32	me	nr
3479	43666	f	ma	80-90	45	nm	re
...	...	...	...	...	...	...	...

order

Customer ID	Order ID	Store ID	Delivery Mode	Paymt Mode
...	...	...	...	...
3478	2140267	12	regular	cash
3478	3446778	12	express	check
3478	4728386	17	regular	check
3479	3233444	17	express	credit
3479	3475886	12	regular	credit
...	...	...	...	...

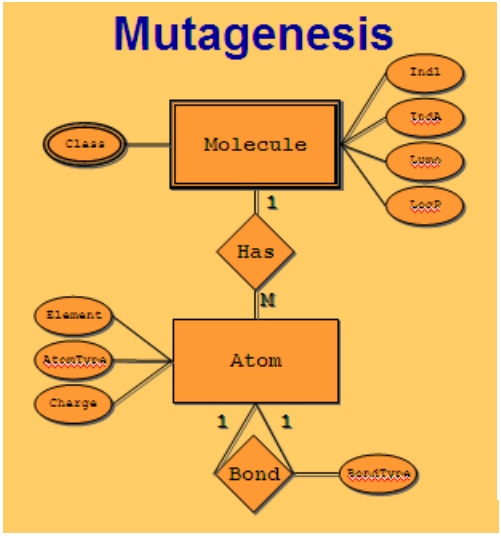
store

Store ID	Size	Type	Location
...	...	...	...
12	small	franchise	city
17	large	indep	rural
...	...	...	...

Relational representation of customers, orders and stores.

Background and motivation: Relational data mining

- **ILP, relational learning, propositionalization**
  - Learning from complex multi-relational data
  - Learning from complex structured data: e.g., molecules and their properties in protein engineering, biochemistry, ...
  - Learning by using domain ontologies (e.g. the gene ontology) as background knowledge for relational data mining

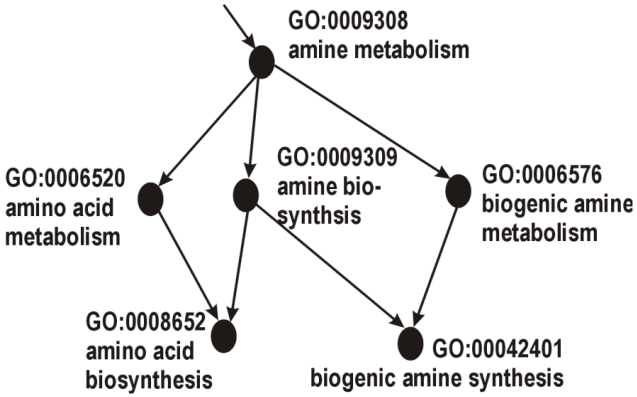


customer							
ID	Zip	Sex	So St	In come	Age	Club	Resp
...	...	...	...	...	...	...	...
3478	34677	m	si	60-70	32	me	nr
3479	43666	f	ma	80-90	45	nm	re
...	...	...	...	...	...	...	...

order				
Customer ID	Order ID	Store ID	Delivery Mode	Paymt Mode
...	...	...	...	...
3478	2140267	12	regular	cash
3478	3446778	12	express	check
3478	4728386	17	regular	check
3479	3233444	17	express	credit
3479	3475886	12	regular	credit
...	...	...	...	...

store			
Store ID	Size	Type	Location
...	...	...	...
12	small	franchise	city
17	large	indep	rural
...	...	...	...

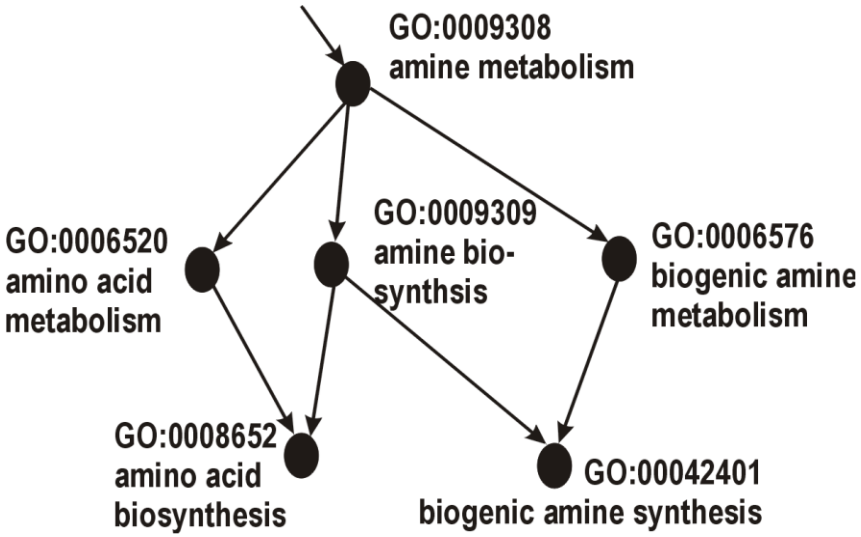
Relational representation of customers, orders and stores.



Background and motivation: Using domain ontologies

Using domain ontologies as background knowledge

- E.g., the Gene Ontology (GO)
- GO is a database of terms, describing gene sets in terms of their
  - functions (12,093)
  - processes (1,812)
  - components (7,459)
- Genes are annotated to GO terms
- Terms are connected (is_a, part_of)
- Levels represent terms generality



Background and motivation: Using domain ontologies

- Using background knowledge in data mining has been a topic of extensive research
  - Hierarchical attribute values, hierarchy/taxonomy of attributes, since 1986
  - ILP, relational data mining, propositionalization, since 1991
  - Ontologies (Tim Berners-Lee), since 1989
    - accepted formalism for consensual knowledge representation for Semantic Web applications, a basic for the Semantic Web
    - Description logic, OWL, Protégé ontology editor
  - Using ontologies in data mining, since 2004

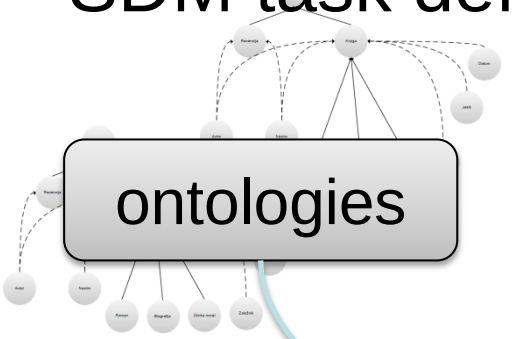
Background and motivation: Early work

- Inducing Multi-Level Association Rules from Multiple Relations (F.A. Lisi and D. Malerba, MLJ 2004)
- Mining the Semantic Web: A Logic-Based Methodology (F.A. Lisi and F. Esposito, ISMIS, 2005)
- using an engineering ontology of CAD elements and structures as BK to extract frequent product design patterns in CAD repositories and discovering predictive rules from CAD data (Zakova et al., ILP 2006)
- using biomedical ontologies as BK in microarray data analysis for finding groups of differentially expressed genes (Zelezny et al., Biomed, 2006)
- Data Mining with Ontologies: Implementations, Findings, and Frameworks, edited by H.O. Nigro, S.G. Cisaro, D. Xodo, Information Science reference, 2008

What is Semantic Data Mining

- Ontology-driven (semantic) data mining is an emerging research topic – the topic of this tutorial
- Semantic Data Mining (SDM) - a new term denoting:
  - the new challenge of mining semantically annotated resources, with ontologies used as background knowledge to data mining
  - approaches with which semantic data are mined

SDM task definition



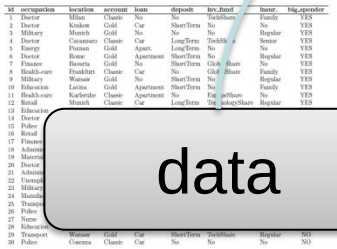
ontologies

annotations, mappings

Semantic data mining

```
target(A) :-
   'Doctor'(A), 'Italy'(A).

target(A)
   'P

targe
   'P                                  'Gold'(A).

targe
   'Germany'(A), 'Insurance'(A).

target(A) :-
   'Service'(A), 'Germany'(A).
```

model, patterns



data

Given:

- transaction data table, relational database, text documents, Web pages, …
- one or more domain ontologies

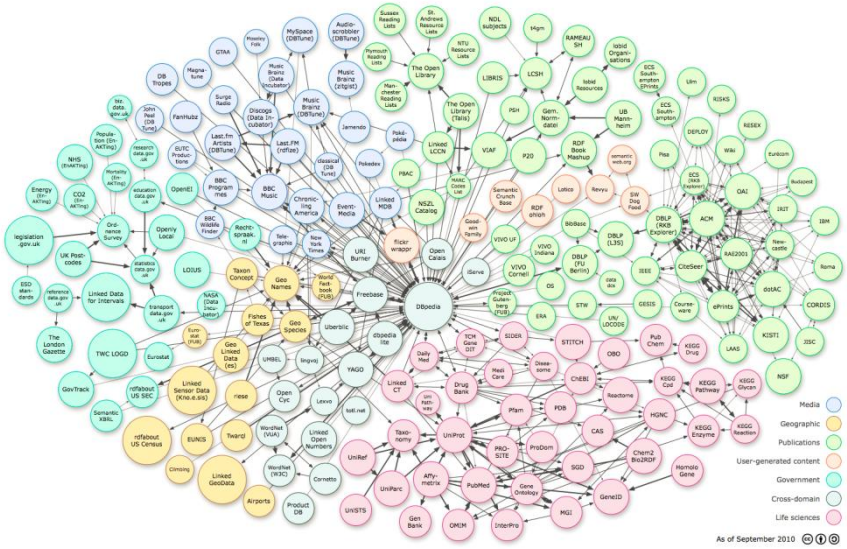
Find: a classification model, a set of patterns

What is Semantic Data Mining

- Current Semantic data mining scenario: Mining empirical data with ontologies as background knowledge
  - abundant empirical data, but
  - scarce background knowledge
- Future Semantic data mining scenario:
  - envisioning a growing amount of semantic data
    - abundance of ontologies and semantically anotated data collections
    - e.g. Linked Data
      - over 6 billion RDF triples
      - over 148 million links

What is Semantic Data Mining

- We may envision a paradigm shift from data mining to knowledge mining
- The envisioned future Semantic data mining scenario in mining the Semantic Web:
  - mining knowledge encoded in domain ontologies,
  - constrained by annotated (empirical) data collections.



What is Semantic Data Mining

- Two different types of semantic resources can be exploited in data mining:
  - Domain ontologies
    - Using domain ontologies as background knowledge (BK) for mining experimental data – see Part 1 of this tutorial
    - Mining OWL ontologies and other annotated resources (DL-learning) – see Part 2
  - Data mining ontologies
    - Developing and using a data mining ontology for meta-mining of data mining workflows – see Part 3

Early work in Semantic subgroup discovery: RSD and SEGS

- Part 1a of this tutorial (N. Lavrac) presents two relational subgroup discovery systems, using domain ontologies as background knowledge in Semantic data mining
  - General purpose system **RSD** for **Relational Subgroup Discovery**, using a propositionalization approach to relational data mining (Zelezny and Lavrac, MLJ 2006)
  - Specialized system **SEGS** for **Searching for Enriched Gene Sets**, performing top-down search of rules, formed as conjunctions of ontology terms (Trajkovski et al., IEEE TSMC 2008, Trajkovski et al., JBI 2008)
- Part 1b of this tutorial (A. Vavpetic) presents **g-SEGS** (2010) and **SDM-Aleph** (2011) by a demo/video

RSD: Propositionalization approach to relational data mining

customer

ID	Zip	S ex	So St	In come	A ge	Cl ub	Re sp
...	...	...	...	...	...	...	...
3478	34677	m	si	60-70	32	me	nr
3479	43666	f	ma	80-90	45	nm	re
...	...	...	...	...	...	...	...

order

Customer ID	Order ID	Store ID	Delivery Mode	Paymt Mode
...	...	...	...	...
3478	2140267	12	regular	cash
3478	3446778	12	express	check
3478	4728386	17	regular	check
3479	3233444	17	express	credit
3479	3475886	12	regular	credit
...	...	...	...	...

store

Store ID	Size	Type	Location
...	...	...	...
12	small	franchise	city
17	large	indep	rural
...	...	...	...

Relational representation of customers, orders and stores.

Step 1

Propositionalization

	f1	f2	f3	f4	f5	f6	...				...	fn
g1	1	0	0	1	1	1	0	0	1	0	1	1
g2	0	1	1	0	1	1	0	0	0	1	1	0
g3	0	1	1	1	0	0	1	1	0	0	0	1
g4	1	1	1	0	1	1	0	0	1	1	1	0
g5	1	1	1	0	0	1	0	1	1	0	1	0
g1	0	0	1	1	0	0	0	1	0	0	0	1
g2	1	1	0	0	1	1	0	1	0	1	1	1
g3	0	0	0	0	1	0	0	1	1	1	0	0
g4	1	0	1	1	1	0	1	0	0	1	0	1

RSD: Propositionalization approach to data mining

customer

ID	Zip	S ex	So St	In come	A ge	Cl ub	Re sp
...	...	...	...	...	...	...	...
3478	34677	m	si	60-70	32	me	nr
3479	43666	f	ma	80-90	45	nm	re
...	...	...	...	...	...	...	...

order

Customer ID	Order ID	Store ID	Delivery Mode	Paymt Mode
...	...	...	...	...
3478	2140267	12	regular	cash
3478	3446778	12	express	check
3478	4728386	17	regular	check
3479	3233444	17	express	credit
3479	3475886	12	regular	credit
...	...	...	...	...

store

Store ID	Size	Type	Location
...	...	...	...
12	small	franchise	city
17	large	indep	rural
...	...	...	...

Relational representation of customers, orders and stores.

Step 1

Propositionalization

1. constructing relational features
2. constructing a propositional table

	f1	f2	f3	f4	f5	f6	...				...	fn
g1	1	0	0	1	1	1	0	0	1	0	1	1
g2	0	1	1	0	1	1	0	0	0	1	1	0
g3	0	1	1	1	0	0	1	1	0	0	0	1
g4	1	1	1	0	1	1	0	0	1	1	1	0
g5	1	1	1	0	0	1	0	1	1	0	1	0
g1	0	0	1	1	0	0	0	1	0	0	0	1
g2	1	1	0	0	1	1	0	1	0	1	1	1
g3	0	0	0	0	1	0	0	1	1	1	0	0
g4	1	0	1	1	1	0	1	0	0	1	0	1

RSD: Propositionalization approach to data mining

customer							
ID	Zip	Sex	So St	In come	Age	Club	Resp
...	...	...	...	...	...	...	...
3478	34677	m	si	60-70	32	me	nr
3479	43666	f	ma	80-90	45	nm	re
...	...	...	...	...	...	...	...

order				
Customer ID	Order ID	Store ID	Delivery Mode	Paymt Mode
...	...	...	...	...
3478	2140267	12	regular	cash
3478	3446778	12	express	check
3478	4728386	17	regular	check
3479	3233444	17	express	credit
3479	3475886	12	regular	credit
...	...	...	...	...

store			
Store ID	Size	Type	Location
...	...	...	...
12	small	franchise	city
17	large	indep	rural
...	...	...	...

Relational representation of customers, orders and stores.

Step 1

Propositionalization

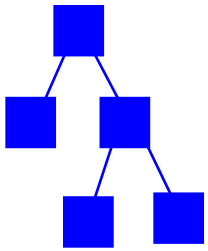
1. constructing relational features
2. constructing a propositional table

	f1	f2	f3	f4	f5	f6	...				...	fn
g1	1	0	0	1	1	1	0	0	1	0	1	1
g2	0	1	1	0	1	1	0	0	0	1	1	0
g3	0	1	1	1	0	0	1	1	0	0	0	1
g4	1	1	1	0	1	1	0	0	1	1	1	0
g5	1	1	1	0	0	1	0	1	1	0	1	0
g1	0	0	1	1	0	0	0	1	0	0	0	1
g2	1	1	0	0	1	1	0	1	0	1	1	1
g3	0	0	0	0	1	0	0	1	1	1	0	0
g4	1	0	1	1	1	0	1	0	0	1	0	1

	f1	f2	f3	f4	f5	f6	...				...	fn
g1	1	0	0	1	1	1	0	0	1	0	1	1
g2	0	1	1	0	1	1	0	0	0	1	1	0
g3	0	1	1	1	0	0	1	1	0	0	0	1
g4	1	1	1	0	1	1	0	0	1	1	1	0
g5	1	1	1	0	0	1	0	1	1	0	1	0
g1	0	0	1	1	0	0	0	1	0	0	0	1
g2	1	1	0	0	1	1	0	1	0	1	1	1
g3	0	0	0	0	1	0	0	1	1	1	0	0
g4	1	0	1	1	1	0	1	0	0	1	0	1

Step 2

Data Mining



model, patterns, …

Relational subgroup discovery with RSD

customer							
ID	Zip	Sex	So St	In come	Age	Club	Resp
...	...	...	...	...	...	...	...
3478	34677	m	si	60-70	32	me	nr
3479	43666	f	ma	80-90	45	nm	re
...	...	...	...	...	...	...	...

order				
Customer ID	Order ID	Store ID	Delivery Mode	Paymt Mode
...	...	...	...	...
3478	2140267	12	regular	cash
3478	3446778	12	express	check
3478	4728386	17	regular	check
3479	3233444	17	express	credit
3479	3475886	12	regular	credit
...	...	...	...	...

store			
Store ID	Size	Type	Location
...	...	...	...
12	small	franchise	city
17	large	indep	rural
...	...	...	...

Relational representation of customers, orders and stores.

Step 1

Propositionalization

1. constructing relational features
2. constructing a propositional table

	f1	f2	f3	f4	f5	f6	...				...	fn
g1	1	0	0	1	1	1	0	0	1	0	1	1
g2	0	1	1	0	1	1	0	0	0	1	1	0
g3	0	1	1	1	0	0	1	1	0	0	0	1
g4	1	1	1	0	1	1	0	0	1	1	1	0
g5	1	1	1	0	0	1	0	1	1	0	1	0
g1	0	0	1	1	0	0	0	1	0	0	0	1
g2	1	1	0	0	1	1	0	1	0	1	1	1
g3	0	0	0	0	1	0	0	1	1	1	0	0
g4	1	0	1	1	1	0	1	0	0	1	0	1

	f1	f2	f3	f4	f5	f6	...				...	fn
g1	1	0	0	1	1	1	0	0	1	0	1	1
g2	0	1	1	0	1	1	0	0	0	1	1	0
g3	0	1	1	1	0	0	1	1	0	0	0	1
g4	1	1	1	0	1	1	0	0	1	1	1	0
g5	1	1	1	0	0	1	0	1	1	0	1	0
g1	0	0	1	1	0	0	0	1	0	0	0	1
g2	1	1	0	0	1	1	0	1	0	1	1	1
g3	0	0	0	0	1	0	0	1	1	1	0	0
g4	1	0	1	1	1	0	1	0	0	1	0	1

Step 2

Subgroup discovery

```
target(A) :-
   'Doctor'(A), 'Italy'(A).

target(A) :-
   'Public'(A), 'Gold'(A).

target(A) :-
   'Poland'(A), 'Deposit'(A), 'Gold'(A).

target(A) :-
   'Germany'(A), 'Insurance'(A).

target(A) :-
   'Service'(A), 'Germany'(A).
```

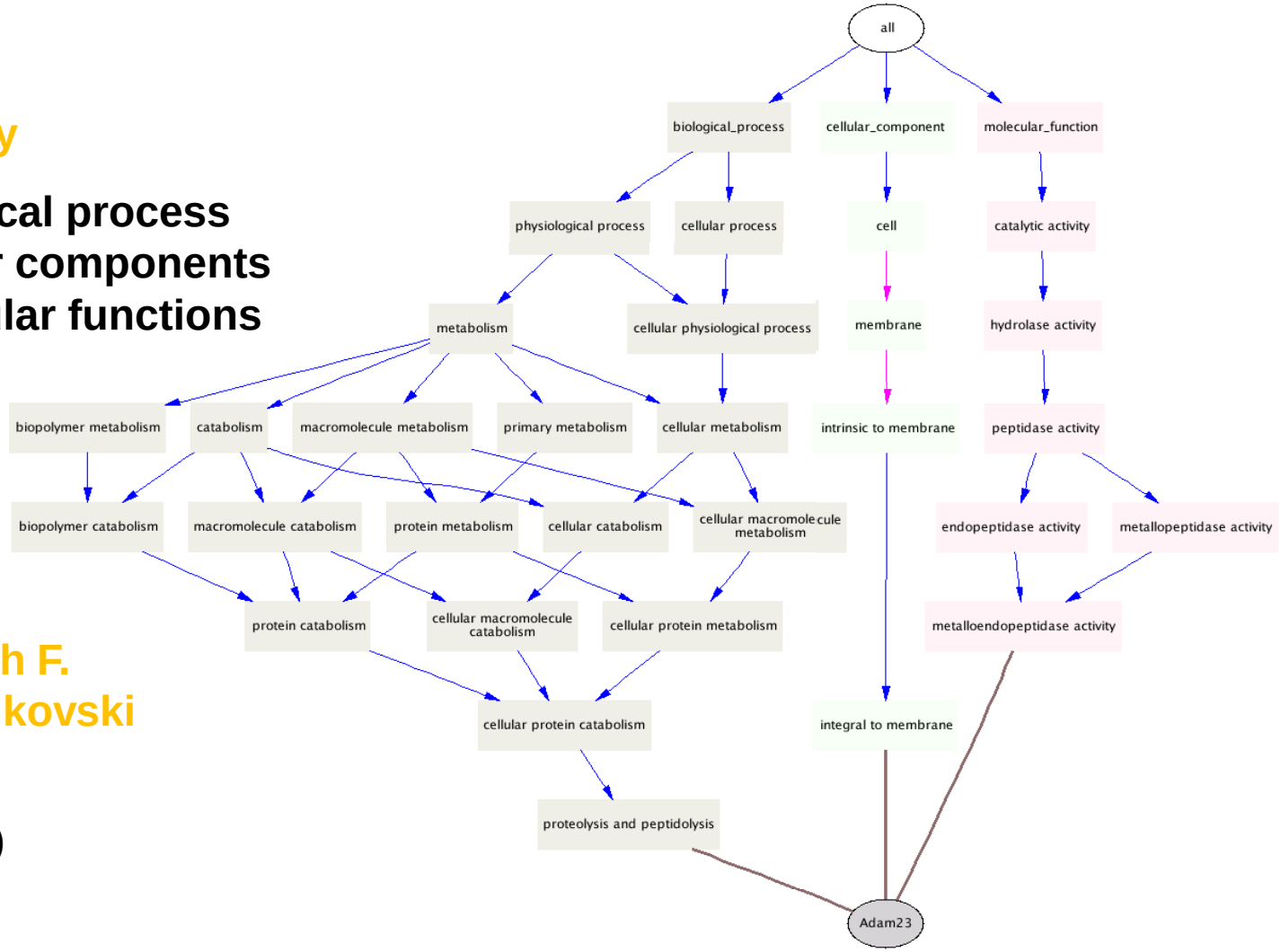
patterns (set of rules)

Semantic subgroup discovery with RSD

Using GO as background knowledge in DNA microarray data analysis with relational subgroup discovery system RSD

Gene Ontology

12,093 biological process
1,812 cellular components
7,459 molecular functions

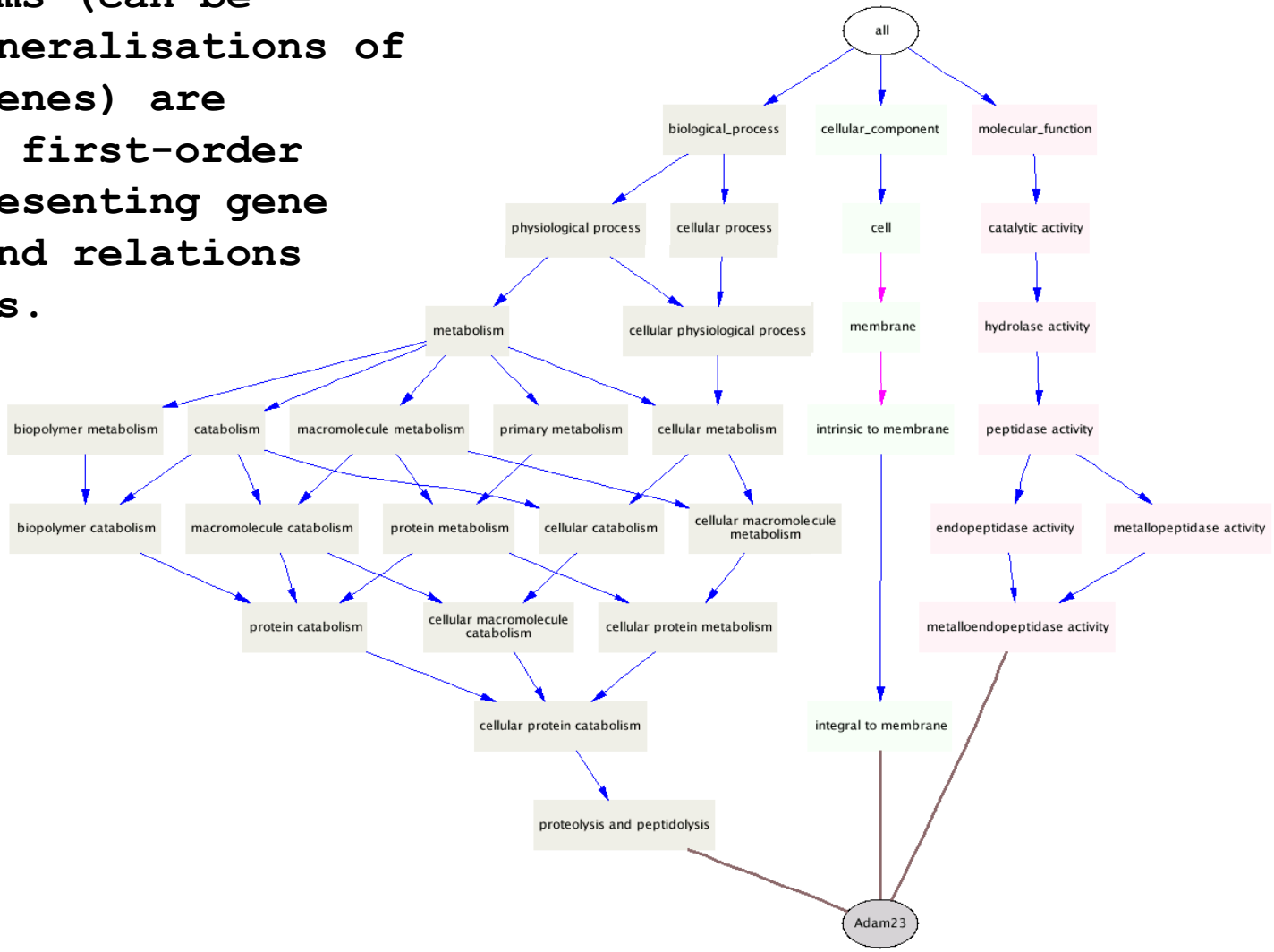


Joint work with F. Zelezny, I. Trajkovski and J. Tolar

(Biomed, 2006)

Semantic subgroup discovery with RSD

Ontology terms (can be viewed as generalisations of individual genes) are described by first-order features, presenting gene properties and relations between genes.



Semantic subgroup discovery with RSD

Application of RSD in microarray data analysis using GO as background knowledge (Zelezny et al., Biomed, 2006)

1. Take ontology terms represented as logical facts, e.g.

```
component(gene2532,'GO:0016020').
function(gene2534,'GO:0030554').
process(gene2534,'GO:0007243').
interaction(gene2534,gene4803).
```

2. Automatically generate generalized relational features:

```
f(2,A):-component(A,'GO:0016020').
f(7,A):-function(A,'GO:0030554').
f(11,A):-process(A,'GO:0007243').
f(224,A):- interaction(A,B),
           function(B,'GO:0016787'),
           component(B,'GO:0043231').
```

3. Propositionalization: Determine truth values of features
4. Learn rules by a subgroup discovery algorithm CN2-SD

Semantic subgroup discovery with RSD

Construction of first order features with support > *min_support*

```
f(7,A):-function(A,'GO:0046872').
f(8,A):-function(A,'GO:0004871').
f(11,A):-process(A,'GO:0007165').
f(14,A):-process(A,'GO:0044267').
f(15,A):-process(A,'GO:0050874').
f(20,A):-function(A,'GO:0004871'), process(A,'GO:0050874').
f(26,A):-component(A,'GO:0016021').
f(29,A):- function(A,'GO:0046872'), component(A,'GO:0016020').
f(122,A):-interaction(A,B),function(B,'GO:0004872').
f(223,A):-interaction(A,B),function(B,'GO:0004871'),
    process(B,'GO:0009613').
f(224,A):-interaction(A,B),function(B,'GO:0016787'),
    component(B,'GO:0043231').
```

existential

RSD: Propositionalization

diffexp g1 (gene64499)
diffexp g2 (gene2534)
diffexp g3 (gene5199)
diffexp g4 (gene1052)
diffexp g5 (gene6036)
….

random g1 (gene7443)
random g2 (gene9221)
random g3 (gene2339)
random g4 (gene9657)
….

	f1	f2	f3	f4	f5	f6	…				…	fn
g1	1	0	0	1	1	1	0	0	1	0	1	1
g2	0	1	1	0	1	1	0	0	0	1	1	0
g3	0	1	1	1	0	0	1	1	0	0	0	1
g4	1	1	1	0	1	1	0	0	1	1	1	0
g5	1	1	1	0	0	1	0	1	1	0	1	0
g1	0	0	1	1	0	0	0	1	0	0	0	1
g2	1	1	0	0	1	1	0	1	0	1	1	1
g3	0	0	0	0	1	0	0	1	1	1	0	0
g4	1	0	1	1	1	0	1	0	0	1	0	1

RSD: Rule construction with CN2-SD

	f1	f2	f3	f4	f5	f6	…				…	fn
g1	1	0	0	1	1	1	0	0	1	0	1	1
g2	0	1	1	0	1	1	0	0	0	1	1	0
g3	0	1	1	1	0	0	1	1	0	0	0	1
g4	1	1	1	0	1	1	0	0	1	1	1	0
g5	1	1	1	0	0	1	0	1	1	0	1	0
g1	0	0	1	1	0	0	0	1	0	0	0	1
g2	1	1	0	0	1	1	0	1	0	1	1	1
g3	0	0	0	0	1	0	0	1	1	1	0	0
g4	1	0	1	1	1	0	1	0	0	1	0	1

Over-expressed

IF

f2 and f3

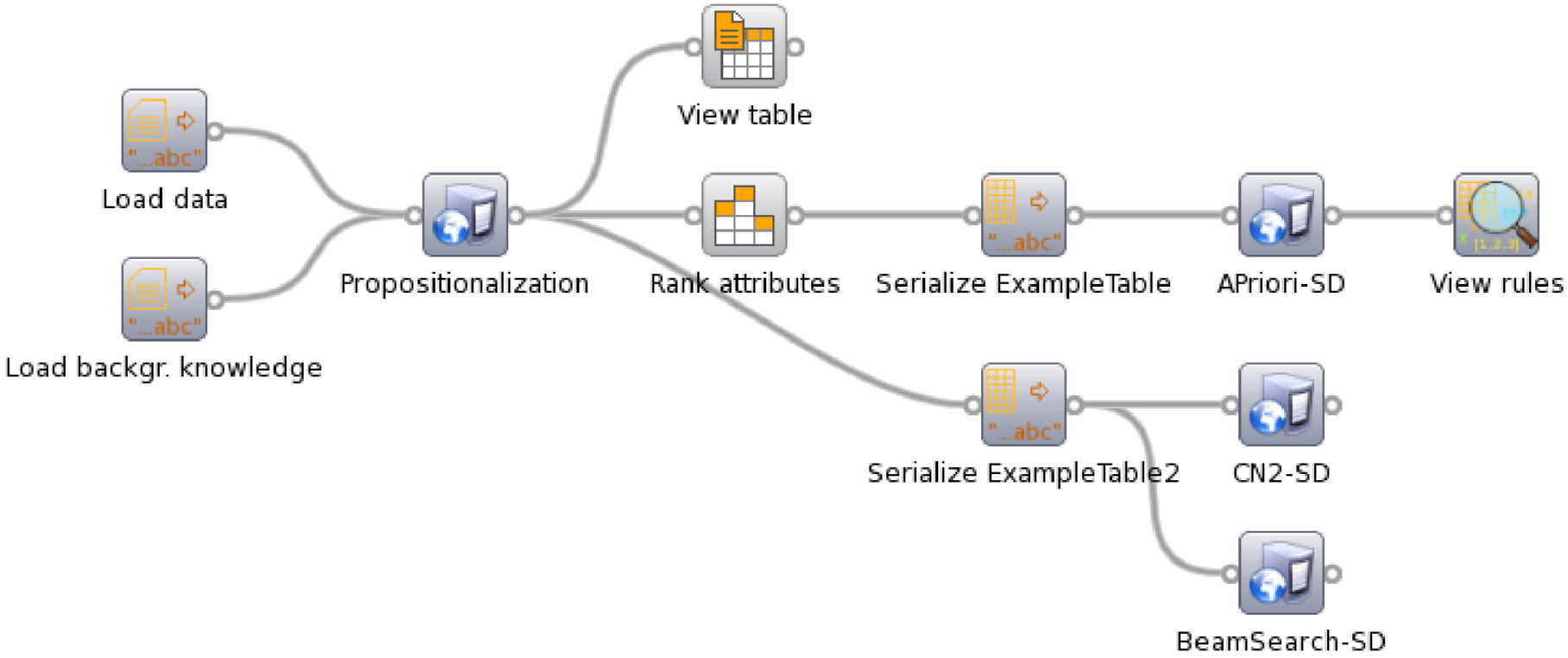
[4,0]

diffexp(A) :- interaction(A,B) & function(B,'GO:0004871')

RSD implementation in Orange4WS

RSD implemented as a workflow in Orange4WS:

- propositionalization
- subgroup discovery algorithms: SD, Apriori-SD, CN2-SD



29

Semantic subgroup discovery with SEGS

- **Gene set enrichment**: moving from single gene to gene set analysis
  - A gene set is enriched if the genes in the set are statistically significantly differentially expressed compared to the rest of the genes.
  - Observation: E.g., an 20% increase in all genes members of a biological pathway may alter the execution of this pathway … and its impact on other processes … significantly more then a 10-fold increase in a single gene.
- System **SEGS** for finding groups of differentially expressed genes from experimental microarray data
- Using biomedical ontologies GO, KEGG and ENTREZ as background knowledge

Semantic subgroup discovery with SEGS

- Gene set enrichment methods:
  - Single GO terms:
    - Gene Set Enrichment Analysis (GSEA)
    - Parametric Analysis of Gene Set Enrichment (PAGE)
  - Conjunctions of GO terms: SEGS
- Results of **Searching for Enriched Gene Sets** with **SEGS**:

  Rules describing groups of genes that are differentially expressed (e.g., belong to class DIFF-EXP of top 300 most differentially expressed genes) in contrast with RANDOM genes (randomly selected genes with low differential expression).
- Sample semantic subgroup description:

```
diffexp(A) :- interaction(A,B) & function(B,'GO:0004871')
              & process(B,'GO:0009613')
```

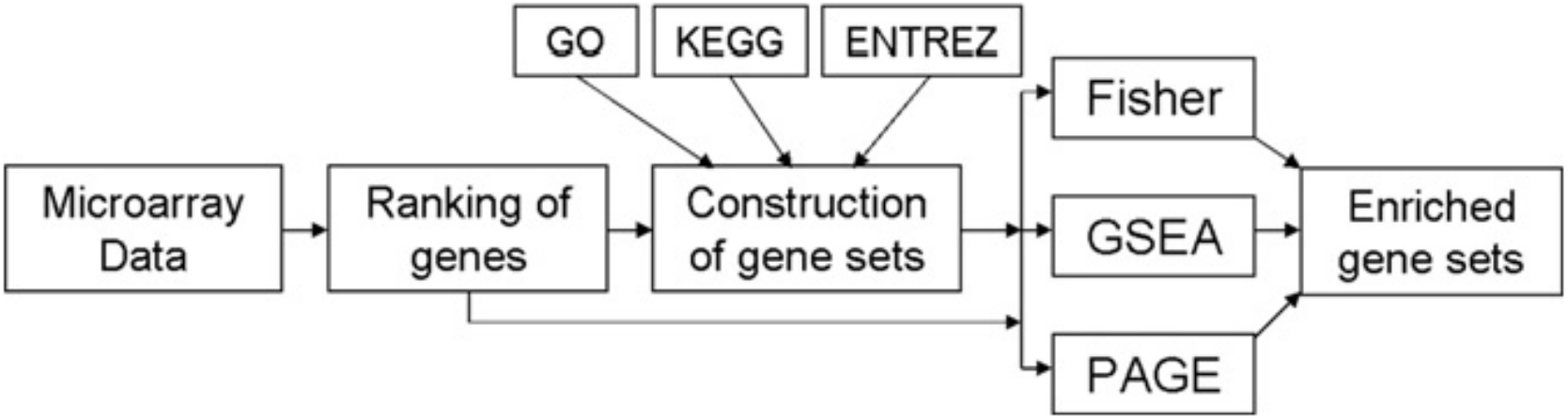
Semantic subgroup discovery with SEGS

The SEGS approach:

- Fuse information from GO, KEGG and ENTREZ
- Generate gene set candidates as conjunctions of GO, KEGG and ENTREZ terms
- Combine Fisher, GSEA and PAGE enrichment tests to select most interesting groups of differentially expressed genes

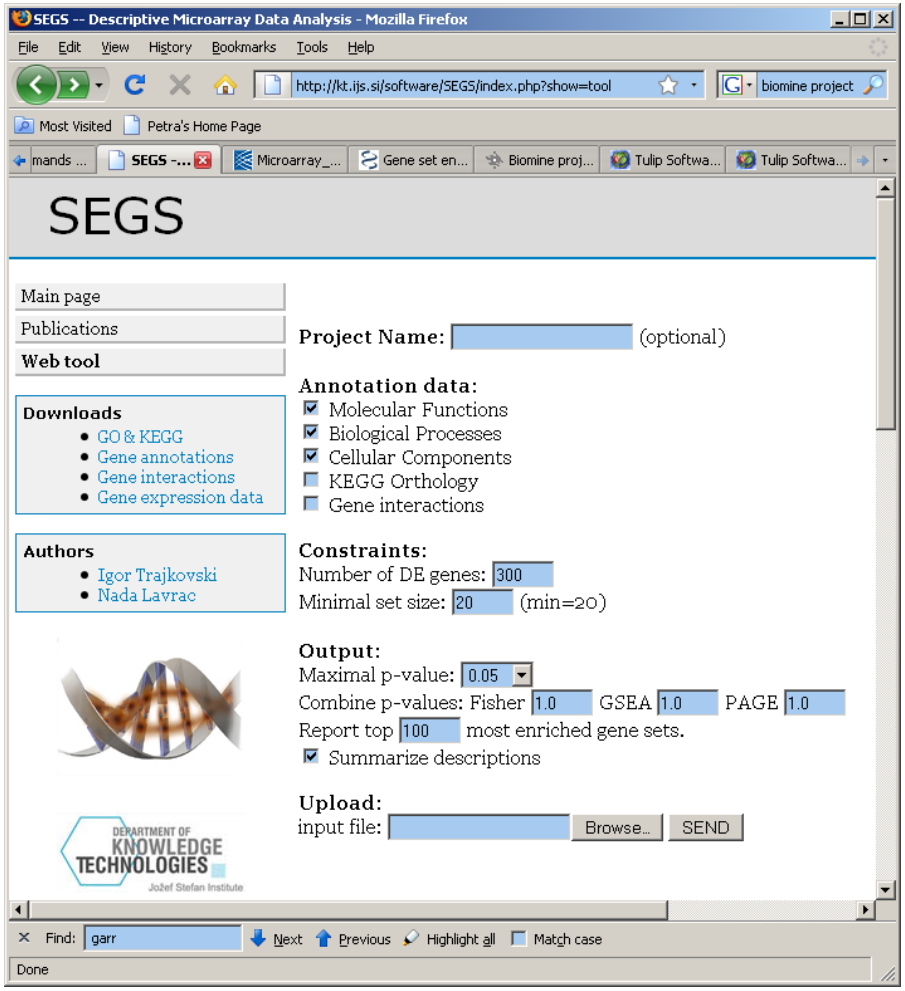
Semantic subgroup discovery with SEGS

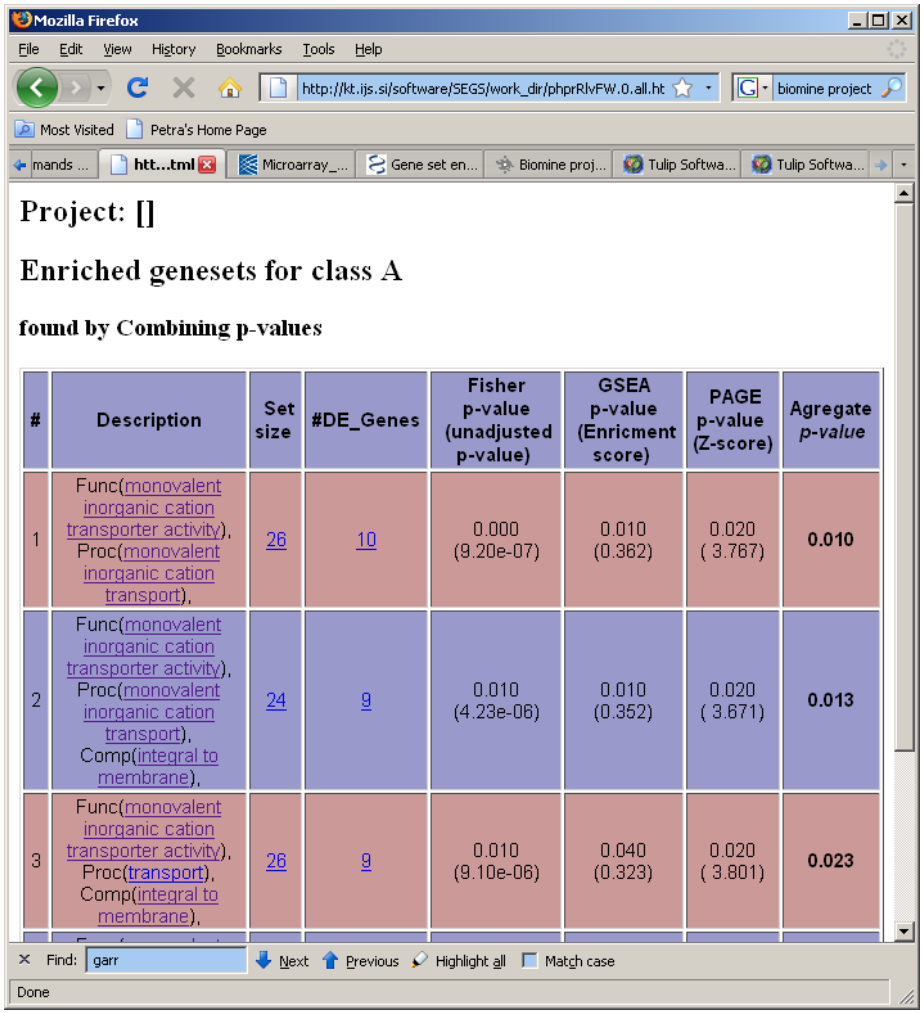
- SEGS workflow is implemented in the Orange4WS data mining environment



- SEGS is also implemented also as a Web applications (Trajkovski et al., IEEE TSMC 2008, Trajkovski et al., JBI 2008)

Semantic subgroup discovery with SEGS

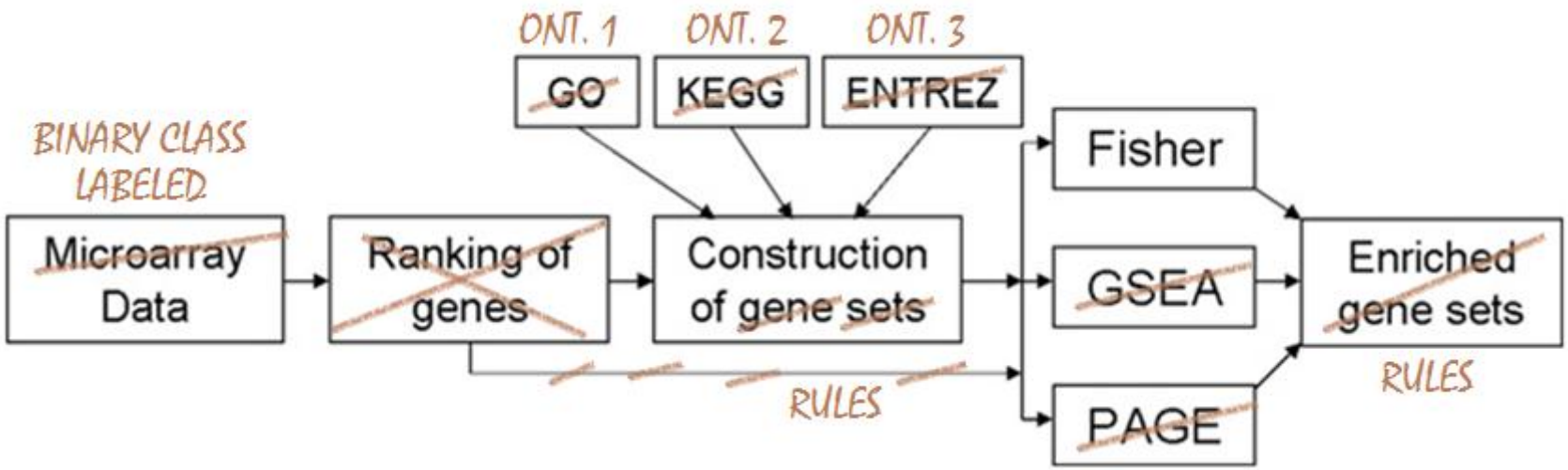




#	Description	Set size	#DE_Genes	Fisher p-value (unadjusted p-value)	GSEA p-value (Enricment score)	PAGE p-value (Z-score)	Agregate *p-value*
1	Func(monovalent inorganic cation transporter activity), Proc(monovalent inorganic cation transport),	26	10	0.000 (9.20e-07)	0.010 (0.362)	0.020 (3.767)	**0.010**
2	Func(monovalent inorganic cation transporter activity), Proc(monovalent inorganic cation transport), Comp(integral to membrane),	24	9	0.010 (4.23e-06)	0.010 (0.352)	0.020 (3.671)	**0.013**
3	Func(monovalent inorganic cation transporter activity), Proc(transport), Comp(integral to membrane),	26	9	0.010 (9.10e-06)	0.040 (0.323)	0.020 (3.801)	**0.023**

From SEGS to g-SEGS: Generalizing SEGS

- g-SEGS: a semantic data mining system generalizing SEGS



- Discovers subgroups both for ranked and labeled data
- Exploits input ontologies in OWL format
- Is also implemented in Orange4WS

Publications in Semantic subgroup discovery

- M. Zakova, F. Zelezny, J.A. Garcia-Sedano, C. Masia Tissot, N. Lavrac, P. Kremen, J. Molina: Relational Data Mining Applied to Virtual Engineering of Product Designs. In Proc. ILP 2006, Springer LNSC 4455, 439-453, 2007.
- I. TRAJKOVSKI, F. ZELEZNY, N. LAVRAC, J. TOLAR: Learning relational destriptions of differentially expressed gene groups. IEEE trans. syst. man cybern., Part C Appl., 2008, vol. 38, no. 1, 16-25.
- I. TRAJKOVSKI, N. LAVRAC, J. TOLAR: SEGS : search for enriched gene sets in microarray data. Journal of biomedical informatics, 2008, vol. 41, no. 4, 588-601.
- Lavrac et al., Semantic subgroup discovery: Using ontologies in microarray data analysis. IEEE EMBC, 2009.
- Podpecan et al. SegMine workflows for semantic microarray data analysis in Orange4WS, Submitted to BMC Bioinformatics, 2011

Other related publications

- Related work on developing/using a data mining ontology for automated data mining workflow composition:
  - 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, vol. 8, no. 2, 253-264, 2011.
  - V. Podpecan, M. Zemenova, and N. Lavrac: Orange4WS Environment for Service-Oriented Data Mining, The Computer Journal, 2011. doi: 10.1093/comjnl/bxr077

Introduction to Semantic Data Mining (SDM)

Nada Lavrac: Part 1a: Introduction

- Background and motivation
- What is Semantic Data Mining: Definition and settings
- Early work in Semantic subgroup discovery

Anže Vavpetič: Part 1.b: Applications and demo

Part 1b Overview

- SDM algorithms
  - g-SEGS
  - SDM-Aleph
- Biomedical applications: comparison on two biological domains
- Demo video
  - Illustrative example
  - Advanced biological use case

g-SEGS

- An SDM system based on SEGS
- Discovers subgroups for labelled or ranked data
- Exploits input OWL ontologies
- Implemented as a web service in Orange4WS
  - Can also be used e.g. in Taverna

g-SEGS: rule construction

- Top-down bounded exhaustive search
  - Enums all rules by taking one concept from each ontology as a conjunct (+ the *interacts* relation)
  - Search space pruning:
    - Exploiting the *subClassOf* relation between concepts
    - Size constraints: min support and max number of rule terms

g-SEGS: rule selection

- The number of generated rules can be large
- Filtering uninteresting and overlapping rules
- wWRAcc:
  - WRAcc using example weights
  - WRAcc was already used in relational subgroup discovery system RSD (Železný and Lavrač, MLJ 2004)

$$wWRAcc(C \leftarrow Cnd) = \frac{n'(Cnd)}{N'} \cdot \left(\frac{n'(Cnd \wedge C)}{n'(Cnd)} - \frac{n'(C)}{N'} \right)$$

  - Ensuring diverse rules which cover different parts of the example space

g-SEGS: rule selection

```
function ruleSelection(examples, k):
   # examples - example set
   # k - the maximum number of rules covering the same example

   # Construct the rule set.
   ruleSet = construct ([], ontologies[0], 0)
   resultSet = []
   repeat
     # Select the currently best rule according to WRAcc.
     rule = bestRule(ruleSet)
     resultSet.add(rule)
     # Decrease weights to covered examples and remove the
     # examples which have been covered k times.
     decreaseWeights(examples, rule, k)
   until examples == [] or ruleSet == []

   # For each rule compute the WRAcc, ignoring the weights.
   for each rule in resultSet:
      rule.score = WRAcc(rule)
   end for
   return resultSet
```

SDM-Aleph

- An SDM system implemented using the popular ILP system Aleph [1]
- Implemented as a WS in Orange4WS
- Same inputs/outputs as g-SEGS
- Any number of additional binary relations

[1] Ashwin Srinivasan
http://www.cs.ox.ac.uk/activities/machlearn/Aleph/aleph.html

SDM-Aleph: rule construction and selection

1. Select example
2. Build a most specific clause for that example (bottom clause)
3. Search: from the bottom clause enumerate all more general clauses which satisfy some conditions (e.g., min support)
4. From the clauses select the best rule according to wracc and add it to the rule set
5. Go to 1

SDM-Aleph: implementation

- For solving similar SDM tasks – convert:
  - Ontologies, examples, example-to-ontology map
- Concept `c`, with child concepts `c1, c2, …, cm`:

```
c(X) :- c1(X) ; c2(X) ; … ; cm(X).
```

- The *k*-th example, annotated by `c1, c2, …, cm`:

```
instance(ik). c1(ik). c2(ik). … cm(ik).
```

- Examples: ranked or labelled
  - Transform into a two-class problem according to a threshold.
- Additional relations:

```
r(i1, i2). % extensional def. of r/2
```

Experimental datasets

- Two publicly available bio microarray datasets
  - ALL (Chiaretti et al., 2004)
  - hMSC (Wagner et al., 2008)
- Gene expression data
  - ALL ~9,000 genes, hMSC ~20,300 genes
- Background knowledge: Gene Ontology and KEGG
- Elaborate preprocessing workflow (designed with biologists) -- see demo

Experimental results

- Comparison with SEGS: less and more diverse rules
- Comparison with Aleph
  - Evaluation: descriptive measures of rule interestingness (Lavrač et al., 2004)
  - Less general and more significant rules, speed

ALL

System	*AvgCov*	*AvgSup*	*AvgSig*	*AvgWRAcc*	*AUC*	*t*[s]
g-SEGS	0.043	0.477	**14.145**	0.0135	0.564	**5.79**
Aleph	**0.108**	**0.967**	6.969	**0.0156**	**0.592**	159.68

hMSC

System	*AvgCov*	*AvgSup*	*AvgSig*	*AvgWRAcc*	*AUC*	*t*[s]
g-SEGS	0.047	0.387	**2.354**	0.0039	**0.530**	**4.40**
Aleph	**0.086**	**0.963**	1.767	**0.0057**	**0.530**	132.53

Example subgroup description

‘RNA binding’ AND ‘ribosome’ AND ‘protein biosynthesis’

or

target(X) :- ‘RNA binding’(X), ‘ribosome’(X), ‘protein biosynthesis’(X)

Demo

- http://kt.ijs.si/anze_vavpetic/SDM/ecml_demo.wmv

- Contact:

{ nada.lavrac, anze.vavpetic }@ijs.si

Learning from Description Logics

Part 2 of the
Tutorial on Semantic Data Mining

Agnieszka Lawrynowicz, Jedrzej Potoniec
Poznan University of Technology

Outline

1. Description logics in a nutshell
2. Learning in description logic - definition
3. DL learning methods and techniques:
   - Concept learning
   - Refinement operators
   - Pattern mining
   - Similarity-based approaches
4. Tools
5. Applications
6. Presentation of a tool: RMonto

Learning in DLs

Definition

Learning in description logics: a machine learning approach that adopts Inductive Logic Programming as the methodology and description logic as the language of data and hypotheses.

Description logics theoretically underpin the state-of-art Web ontology representation language, **OWL**, so description logic learning approaches are well suited for semantic data mining.

Description logic

Definition

Description Logics, $\mathcal{DL}s$ = family of first order logic-based formalisms suitable for representing knowledge, especially terminologies, ontologies.

Description logic

Definition

Description Logics, $\mathcal{DL}s$ = family of first order logic-based formalisms suitable for representing knowledge, especially terminologies, ontologies.

- subset of first order logic (decidability, efficiency, expressivity)
- root: semantic networks, frames

Basic building blocks $\mathcal{DL}$

- concepts
- roles
- constructors
- individuals

Examples

Atomic concepts: **Artist**, **Movie**

Role: **creates**

Constructors: $\sqcap$, $\exists$

Concept definition: **Director** $\equiv$ **Artist** $\sqcap$ $\exists$**creates.Movie**

Axiom ("each director is an artist"): **Director** $\sqsubseteq$ **Artist**

Asertion: **creates(sofiaCoppola, lostInTranslation)**

$\mathcal{DL}$ knowledge base

$\mathcal{K} = (\mathcal{T}Box, \mathcal{A}Box)$

$\mathcal{T}Box$ = {

CreteHolidaysOffer $\equiv$ Offer $\sqcap\exists$ in.Crete $\sqcap\forall$ in.Crete

SantoriniHolidaysOffer $\equiv$ Offer $\sqcap\exists$ in.Santorini $\sqcap\forall$ in.Santorini

TromsøyaHolidaysOffer $\equiv$ Offer $\sqcap\exists$ in.Tromsøya $\sqcap\forall$ in.Tromsøya

Crete $\sqsubseteq \exists$ partOf.Greece

Santorini $\sqsubseteq \exists$ partOf.Greece

Tromsøya $\sqsubseteq \exists$ partOf.Norway }.

$\mathcal{A}Box$ = {

Offer(o1). in(Crete).

SantoriniHolidaysOffer(o2).

Offer(o3). in(Santorini). hasPrice(o3, 300)

}.

$\mathcal{DL}$ reasoning services

- satisfiability
- inconsistency
- subsumption
- instance checking

Concept learning

Given

- new target concept name C
- knowledge base $\mathcal{K}$ as background knowledge
- a set E^+ of positive examples, and a set E^- of negative examples

the goal is to learn a concept definition $C \equiv D$ such that
$\mathcal{K} \cup \{C \equiv D\} \models E^+$ and $\mathcal{K} \cup \{C \equiv D\} \models E^-$

Negative examples and Open World Assumption

But what are negative examples in the context of the Open World Assumption?

Semantics: ”closed world” vs ”open world”

- Closed world (Logic programming LP, databases)
  - *complete* knowledge of instances
  - lack of information is by default negative information (*negation-as-failure*)

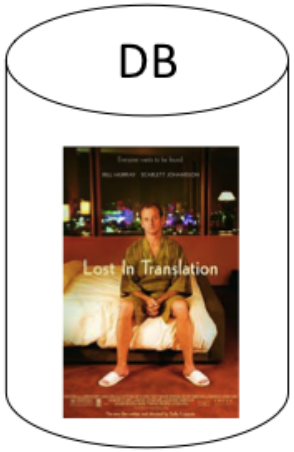
Semantics: ”closed world” vs ”open world”

- Closed world (Logic programming LP, databases)
  - *complete* knowledge of instances
  - lack of information is by default negative information (*negation-as-failure*)
- Open world (description logic DL, Semantic Web)
  - *incomplete* knowledge of instances
  - negation of some fact has to be explicitely asserted (*monotonic negation*)

”Closed world” vs ”open world” example

Let data base contain the following data:

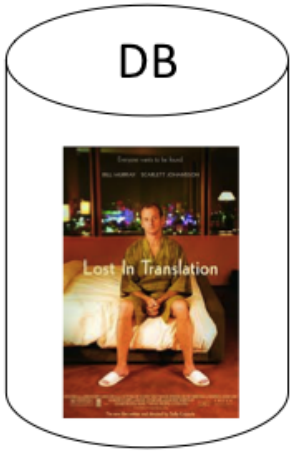
OscarMovie(lostInTranslation)
Director(sofiaCoppola)
creates(sofiaCoppola, lostInTranslation)



”Closed world” vs ”open world” example

Let data base contain the following data:

OscarMovie(lostInTranslation)
Director(sofiaCoppola)
creates(sofiaCoppola, lostInTranslation)

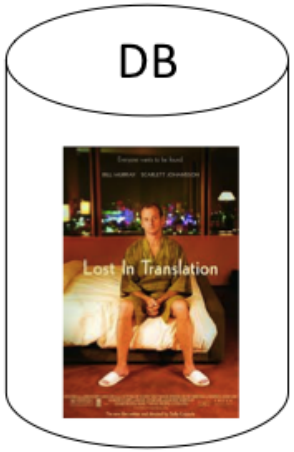


Are all of the movies of Sofia Coppola Oscar movies?

”Closed world” vs ”open world” example

Let data base contain the following data:

OscarMovie(lostInTranslation)
Director(sofiaCoppola)
creates(sofiaCoppola, lostInTranslation)



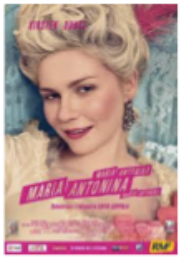
Are all of the movies of Sofia Coppola Oscar movies?

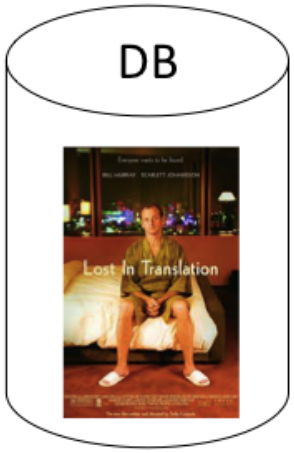
YES - closed world

”Closed world” vs ”open world” example

Let data base contain the following data:

OscarMovie(lostInTranslation)
Director(sofiaCoppola)
creates(sofiaCoppola, lostInTranslation)





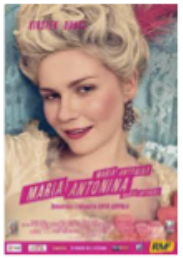
Are all of the movies of Sofia Coppola Oscar movies?

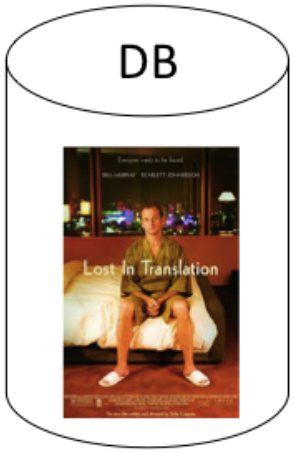
YES - closed world DON’T KNOW - open world

”Closed world” vs ”open world” example

Let data base contain the following data:

OscarMovie(lostInTranslation)
Director(sofiaCoppola)
creates(sofiaCoppola, lostInTranslation)





Are all of the movies of Sofia Coppola Oscar movies?

YES - closed world DON’T KNOW - open world

Different conclusions!

OWA and machine learning

OWA is problematic for machine learning since an individual is rarely deduced to belong to a complement of a concept unless explicitely asserted so.

Dealing with OWA in learning

Solution1: alternative problem setting

Solution2: **K** operator

Solution3: new performance measures

Dealing with OWA in learning: alternative problem setting

”Closing” the knowledge base to allow performing instance checks under the Closed World Assumption (CWA).

By default:
Positive examples of the form $C(a)$, and negative examples of the form $\neg C(a)$, where a is an individual and holding:
$\mathcal{K} \cup \{C \equiv D\} \models E^+$ and $\mathcal{K} \cup \{C \equiv D\} \models E^-$

Alternatively:
Examples of the form $C(a)$ and holding: $\mathcal{K} \cup \{C \equiv D\} \models E^+$ and $\mathcal{K} \cup \{C \equiv D\} \not\models E^-$

Dealing with OWA in learning: **K** operator

- epistemic **K**–operator allows for querying for known properties of known individuals w.r.t. the given knowlege base $\mathcal{K}$
- the **K** operator alters constructs like $\forall$ in a way that they operate on a Closed World Assumption.

Consider two queries:
Q1: $\mathcal{K} \models \{(\forall \text{creates.OscarMovie})\ (\text{sofiaCoppola})\}$
Q2: $\mathcal{K} \models \{(\forall \mathbf{K}\text{creates.OscarMovie})\ (\text{sofiaCoppola})\}$

- Badea and Nienhuys-Cheng (ILP 2000) considered the **K** operator from a theoretical point of view.
- not easy to implement in reasoning systems, non-standard

Dealing with OWA in learning: new performance measures

d'Amato et al (ESWC 2008)
– overcoming unknown answers from the reasoner (as a reference system)
– correspondence between the classification by the reasoner for the instances w.r.t. the test concept C and the definition induced by a learning system

- match rate: number of individuals with exactly the same classification by both the inductive and the deductive classifier w.r.t the overall number of individuals;
- omission error rate: number of individuals not classified by inductive method, relevant to the query w.r.t. the reasoner;
- commission error rate: number of individuals found relevant to C, while they (logically) belong to its negation or vice-versa;
- induction rate: number of individuals found relevant to C or to its negation, while either case not logically derivable from $\mathcal{K}$;

Concept learning - algorithms

supervised:

- YINYANG (Iannone et al, Applied Intelligence 2007)
- DL-Learner (Lehmann & Hitzler, ILP 2007)
- DL-FOIL (Fanizzi et al, ILP 2008)
- TERMITIS (Fanizzi et al, ECML/PKDD 2010)

unsupervised:

- KLUSTER (Kietz & Morik, MLJ 1994)

DL-learning as search

- learning in DLs can be seen as search in space of concepts
- it is possible to impose ordering on this search space using subsumption as natural **quasi-order**, and **generality measure** between concepts
  - if $D \sqsubseteq C$ then C covers all instances that are covered by D
- refinement operators may be applied to traverse the space by computing a set of specializations (resp. generalizations) of a concept

Properties of refinement operators

Consider **downward refinement operator** ρ, and by $C \rightsquigarrow_\rho D$ denote a refinement chain from a concept C to D

- complete: each point in lattice is reachable (for $D \sqsubseteq C$ there exists E such that $E \equiv D$ and a refinement chain $C \rightsquigarrow_\rho \ldots \rightsquigarrow_\rho E$
- weakly complete: for any concept C with $C \sqsubseteq \top$, concept E with $E \equiv C$ can be reached from $\top$
- finite: finite for any concept
- redundant: there exist two different refinement chains from C to D
- proper: $C \rightsquigarrow_\rho D$ implies $C \not\equiv D$

ideal = complete + proper + finite

Combining properties

Can an operator have all of these properties?
Which properties can be combined?

Refinement operators - property theorem

Lehmann & Hitzler (ILP 2007, MLJ 2010) proved that for many DLs, even simpler then those underpinning OWL, **no ideal refinement operator** exists:

learning in DLs is hard

Maximal sets of properties of $\mathcal{L}$ refinement operators which can be combined for $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCN}, \mathcal{SHOIN}, \mathcal{SROIQ}\}$:

1. {weakly complete, complete, finite}
2. {weakly complete, complete, proper}
3. {weakly complete, non-redundant, finite}
4. {weakly complete, non-redundant, proper}
5. {non-redundant, finite, proper}

Pattern mining

Pattern = recurring structure

Data

Pattern

itemsets, sequences, graphs, clauses,...

Patterns in DLs

How to represent patterns in learning from DLs?

Frequent DL concept mining

Lawrynowicz & Potoniec (ISMIS 2011)

- Fr-ONT: mining frequent patterns, where a pattern is in the form of $\mathcal{EL}^{++}$ concept C
- each C is subsumed by a reference concept $\hat{C}$ ($C \sqsubseteq \hat{C}$)
- support calculated as the ratio between the number of instances of C and $\hat{C}$ in $\mathcal{K}$

Example pattern:

$\hat{C}$ = Offer

C = Offer $\sqcap \exists$in.Santorini

$$support(C, \hat{C}, KB) = \frac{2}{3}$$

Clustering in DLs

Classically:

- objects represented as feature vectors in an n-dimensional space
- features may be of different types, but many algorithms are designed to cluster interval-based (numerical) data
  - such algorithms may employ centroid to represent a cluster

DLs:

- individuals in DL knowledge bases are objects to be clustered
- DL individuals need to be logically manipulated
- similarity measures for DLs need to be defined
- DL specific cluster representative may be necessary

(Dis)-similarity measures for DLs

- Language-dependent
  - structural, intensional: decompose concepts structurally, and try to assess an overlap function for each construtor of the considered logic, then aggregate the results of the overlap functions
  - a new measure has to be defined for each logic, this does not easily scale to more expressive DLs
- Language-independent
  - extensional: based on the ABox, checking individual membership to concepts

Language-dependent measures

- simple DL, allowing only disjunction (Borgida et al., 2005)
- $\mathcal{ALC}$ (d'Amato et al., 2005, SAC 2006)
- $\mathcal{ALCNR}$ (Janowicz 2006)
- $\mathcal{EL}^{++}$ (Jozefowski et al, COLISD at ECML/PKDD 2011)

Language-independent measures: example

(Fanizzi et al. DL 2007)

- basic idea inspired by (Sebag 1997): individuals compared on the grounds of their behavior w.r.t. a set of discriminating features
- on a semantic level, similar individuals should behave similarly w.r.t. the same concepts
- $F = F_1, F_2, ..., F_m$ - a collection of (primitive or defined) concept descriptions
- checking whether an individual belongs to F_i, $\neg F_i$ or none of them
- aggregating the results in a way inspired to Minkowski's norms L_p

Semantic similarity measure

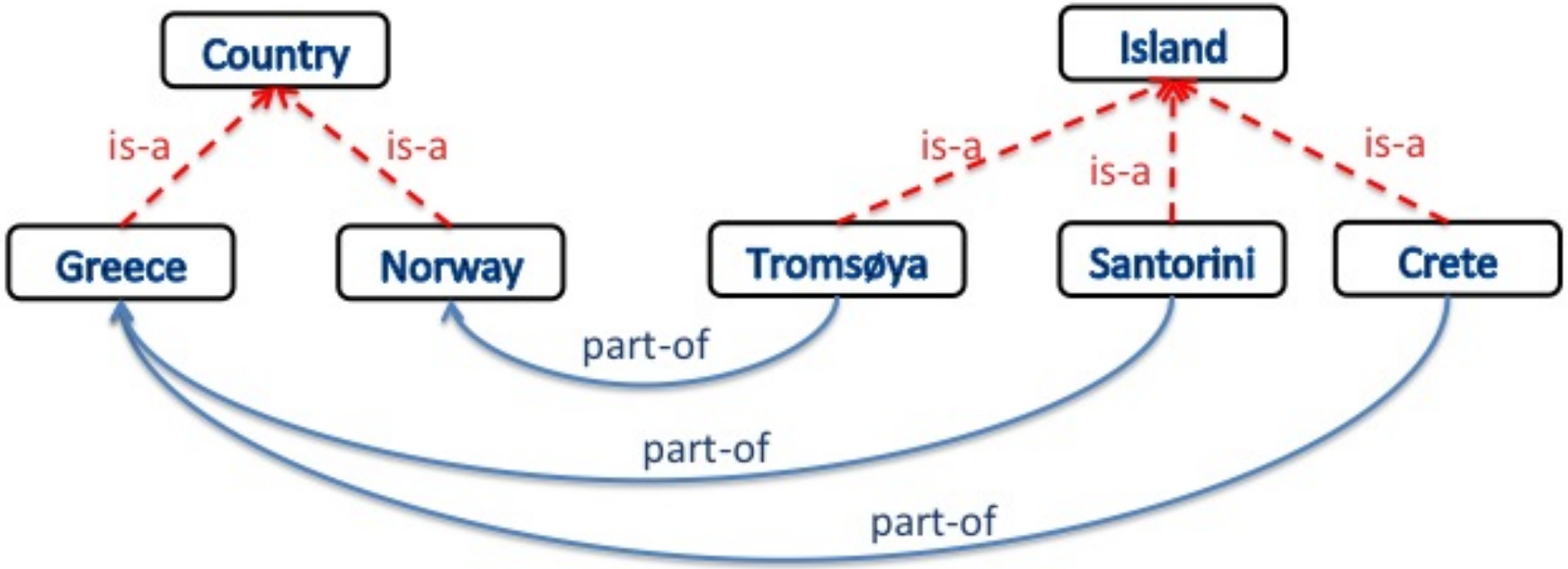
But what is a truly ”semantic” similarity measure?

Semantic similarity measure properties

d'Amato et al. (EKAW 2008) formalized a set of criteria for a measure to satisfy for correctly handling ontological representations:

- soundness: ability to take the semantics of $\mathcal{K}$ (e.g. subsumption hierarchy) into account
- equivalence soundness: ability to recognize semantically equivalent concepts as equal w.r.t. the given measure
- disjointness compatibility: ability to recognize similarities between disjoint concepts

Semantic similarity measure properties - example



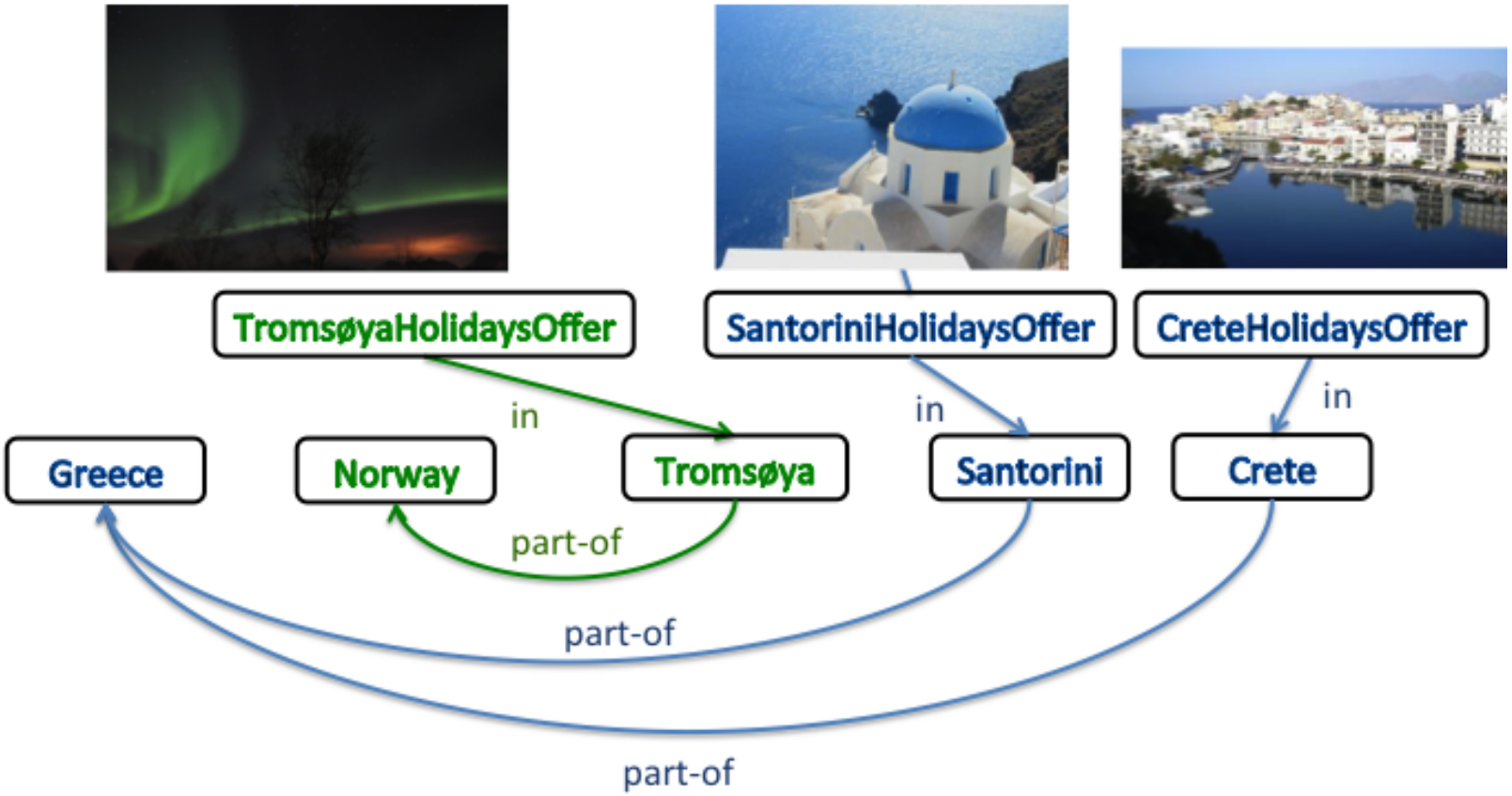
$\text{CreteHolidaysOffer} \equiv \text{Offer} \sqcap \exists\ \text{in.Crete} \sqcap \forall\ \text{in.Crete}$

$\text{SantoriniHolidaysOffer} \equiv \text{Offer} \sqcap \exists\ \text{in.Santorini} \sqcap \forall\ \text{in.Santorini}$

$\text{TromsøyaHolidaysOffer} \equiv \text{Offer} \sqcap \exists\ \text{in.Tromsøya} \sqcap \forall\ \text{in.Tromsøya}$

Soundness

CreteHolidaysOffer should be assesed more similar to **SantoriniHolidaysOffer** than to **TromsøyaHolidaysOffer** since both are located in Greece



Equivalence soundness

Let us assume there exist two concept definitions:
SantoriniHolidaysOffer $\equiv$ Offer $\sqcap\exists$ in.Santorini $\sqcap\forall$ in.Santorini
ThiraHolidaysOffer $\equiv$ Offer $\sqcap\exists$ in.Santorini $\sqcap\forall$ in.Santorini

Since concept names **SantoriniHolidaysOffer** and **ThiraHolidaysOffer** represent semantically equivalent concepts, it should hold:

sim(SantoriniHolidaysOffer, TromsøyaHolidaysOffer) = sim(ThiraHolidaysOffer, TromsøyaHolidaysOffer)

Santorini (officially **Thira)**

Disjointness compatibility

Let us assume we assert in $\mathcal{K}$:
SantoriniHolidaysOffer $\equiv \neg$ CreteHolidaysOffer

This should not necessarily mean the offers are totally different.

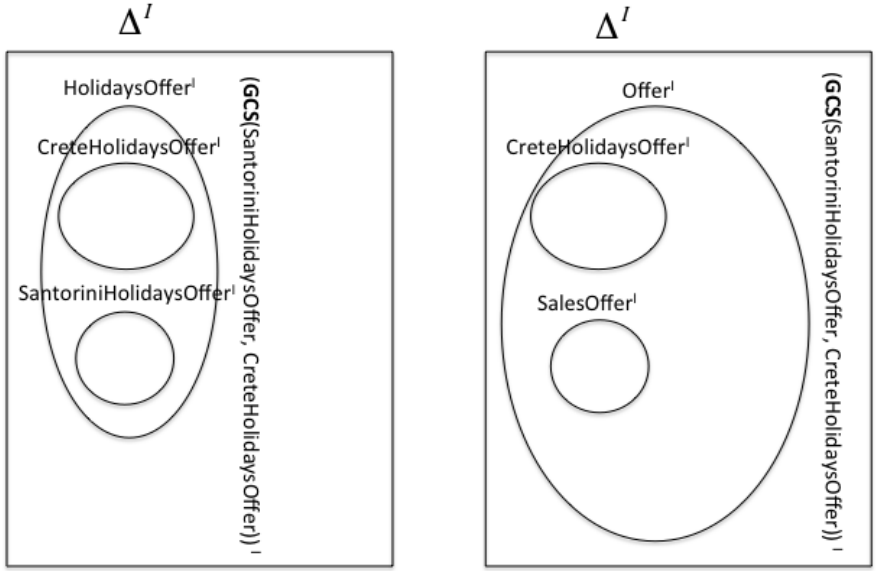
They both represented offers located in Greece, and thus have more commonalities then arbitrary offers. That's why it should hold:

sim(SantoriniHolidaysOffer, CreteHolidaysOffer) >
sim(SantoriniHolidaysOffer, Offer)

GCS-based semantic measure

d'Amato et al. (EKAW 2008)

- many of the "traditional" measures when applied to DLs, and also DL-specific measures fail to meet these semantic criteria
- "semantic" measure based on common super-concept (Good Common Subsumer, GCS of the concepts)
- two concepts are more similar as much their extensions are similar



Problem: GCS not defined for most expressive DLs

DL Learning: available tools

- YINYANG , University of Bari, Iannone 2006
- DL-Learner, University of Leipzig, Lehmann 2006
- RMonto, Poznan University of Technology, Potoniec & Lawrynowicz 2011

DL Learning: applications

- ontology learning, refinement, e.g. d'Amato et al. SWJ 2010, Lehmann et al., ISWC 2010, J. Web. Sem 2011
- service (e.g. semantic Web service) retrieval, e.g. d'Amato et al, IJSC 2010
- semantic aggregation of query results, e.g. Lawrynowicz et al. ICCCI 2009, 2011
- ILP style applications with ontologies

What is RapidMiner?

From RapidMiner brochure

RapidMiner is fully integrated platform for Data Mining, Predictive Analytics and Bussiness Inteligence:

- Rapid Prototyping and Beyond: from the first explorative analysis to the production-ready solution in a few steps;
- Intelligent Bussiness Intelligence: ETL, OLAP, Predictive Modeling, and Reporting combined in a single solution from a single vendor;
- Easy Connections: numerous connectors for all common data bases and data formats as well as unstructured data like text documents;
- Modular System: maximal flexibility and easily extendible.

What we provide?

RMonto

- RapidMiner 5 extension;
- flexible replacing a reasoning tool;
- loading data from heterogeneous sources;

Installation

Visit our website at `http://semantic.cs.put.poznan.pl/RMonto/` and:

1. Download JAR file with RMonto and put it into $RAPIDMINER_HOME/lib/plugins.
2. Download JAR file(s) with one or more PutOntoAPI plugins and put it anywhere inside $RAPIDMINER_HOME.
3. Download (from other websites) reasoning software and put it anywhere inside $RAPIDMINER_HOME keeping files named as specified at our website.

Supported operations

- loading data from files and SPARQL endpoints;

Supported operations

- loading data from files and SPARQL endpoints;
- reasoning with Pellet or Sesame/OWLim;

Supported operations

- loading data from files and SPARQL endpoints;
- reasoning with Pellet or Sesame/OWLim;
- constructing list of learning examples based on KB;

Supported operations

- loading data from files and SPARQL endpoints;
- reasoning with Pellet or Sesame/OWLim;
- constructing list of learning examples based on KB;
- constructing features from KB TBox;

Supported operations

- loading data from files and SPARQL endpoints;
- reasoning with Pellet or Sesame/OWLim;
- constructing list of learning examples based on KB;
- constructing features from KB TBox;
- calculating similarity between individuals;

Supported operations

- loading data from files and SPARQL endpoints;
- reasoning with Pellet or Sesame/OWLim;
- constructing list of learning examples based on KB;
- constructing features from KB TBox;
- calculating similarity between individuals;
- semantic-aware clustering;

Supported operations

- loading data from files and SPARQL endpoints;
- reasoning with Pellet or Sesame/OWLim;
- constructing list of learning examples based on KB;
- constructing features from KB TBox;
- calculating similarity between individuals;
- semantic-aware clustering;
- frequent pattern mining;

Supported operations

- loading data from files and SPARQL endpoints;
- reasoning with Pellet or Sesame/OWLim;
- constructing list of learning examples based on KB;
- constructing features from KB TBox;
- calculating similarity between individuals;
- semantic-aware clustering;
- frequent pattern mining;
- data transformation: propositionalisation;

Acknowledgements

Some presentation ideas inspired on/borrowed from: Claudia d'Amato, Nicola Fanizzi, Jens Lehmann

Bibliography I

[1] L. Badea and S-H. Nienhuys-Cheng. A refinement operator for description logics. In James Cussens and Alan M. Frisch, editors, ILP, volume 1866 of Lecture Notes in Computer Science, pages 40-59. Springer, 2000.

[2] S. Bloehdorn and Y. Sure. Kernel methods for mining instance data in ontologies. In K. Aberer and et al., editors, Proceedings of the 6th International Semantic Web Conference, ISWC2007, volume 4825 of LNCS, pages 58-71. Springer, 2007.

[3] A. Borgida, T.J. Walsh, and H. Hirsh. Towards measuring similarity in description logics. In I. Horrocks, U. Sattler, and F.Wolter, editors, Working Notes of the International Description Logics Workshop, volume 147 of CEUR Workshop Proceedings. CEUR, Edinburgh, UK, 2005.

[4] C. d'Amato, N. Fanizzi, and F. Esposito. A dissimilarity measure for ALC concept descriptions. In Proceedings of the 21st Annual ACM Symposium of Applied Computing, SAC2006, volume 2, pages 1695-1699, Dijon, France, 2006. ACM.

[5] C. d'Amato, N. Fanizzi, and F. Esposito. Query answering and ontology population: An inductive approach. In S. Bechhofer and et al., editors, Proceedings of the 5th European Semantic Web Conference, ESWC2008, volume 5021 of LNCS, pages 288-302. Springer, 2008.

[6] C. d'Amato, S.; Staab, and N. Fanizzi. On the influence of description logics ontologies on conceptual similarity. In A. Gangemi and J. Euzenat, editors, Proceedings of the 16th EKAW Conference, EKAW2008, volume 5268 of LNAI, pages 48-63. Springer, 2008.

[7] C. d'Amato, S. Staab, N. Fanizzi, F. Esposito: Dl-Link: a Conceptual Clustering Algorithm for Indexing Description Logics Knowledge Bases. Int. J. Semantic Computing 4(4): 453-486 (2010)

[8] C. d'Amato, N. Fanizzi, F. Esposito: Inductive learning for the Semantic Web: What does it buy? Semantic Web 1(1-2): 53-59 (2010)

[9] N. Fanizzi, C. d'Amato, and F. Esposito. DL-Foil: Concept learning in Description Logics. In F. Zelezny and N. Lavrac, editors, Proceedings of the 18th International Conference on Inductive Logic Programming, ILP2008, volume 5194 of LNAI, pages 107-121. Springer, Prague, Czech Rep., 2008.

[10] N. Fanizzi, C. d'Amato, and Floriana Esposito. Induction of concepts in web ontologies through terminological decision trees. In Jose L. Balcazar, Francesco Bonchi, Aristides Gionis, and Michele Sebag, editors, ECML/PKDD (1), volume 6321 of Lecture Notes in Computer Science, pages 442-457. Springer, 2010.

[11] L. Iannone, I. Palmisano, and N. Fanizzi. An algorithm based on counterfactuals for concept learning in the Semantic Web. Appl. Intell., 26(2):139-159, 2007.

Bibliography II

[12] K. Janowicz: Sim-DL: Towards a Semantic Similarity Measurement Theory for the Description Logic ALCNR in Geographic Information Retrieval. OTM Workshops (2) 2006: 1681-1692

[13] L. Jozefowski, A. Lawrynowicz, J. Jozefowska, J. Potoniec, and T. Lukaszewski: Kernels for measuring similarity of EL++ description logic concepts, COLISD 2011 workshop at ECML/PKDD 2011.

[14] J.-U. Kietz and K. Morik. A polynomial approach to the constructive induction of structural knowledge. Machine Learning, 14(2):193-218, 1994.

[15] A. Lawrynowicz. Grouping results of queries to ontological knowledge bases by conceptual clustering. In Ngoc Thanh Nguyen, Ryszard Kowalczyk, and Shyi-Ming Chen, editors, ICCCI, volume 5796 of Lecture Notes in Computer Science, pages 504-515. Springer, 2009.

[16] A. Lawrynowicz, J. Potoniec, Fr-ONT: An Algorithm for Frequent Concept Mining with Formal Ontologies, In Proc. of 19th International Symposium on Methodologies for Intelligent Systems (ISMIS 2011), Warsaw, Poland, LNAI, Springer-Verlag, 2011

[17] A. Lawrynowicz, J. Potoniec, L. Konieczny, M. Madziar, A. Nowak, K. Pawlak: ASPARAGUS - a system for automatic SPARQL query results aggregation using semantics, ICCCI 2011, LNCS, Spinger.

[18] J. Lehmann. DL-learner: Learning concepts in description logics. Journal of Machine Learning Research (JMLR), 10:2639-2642, 2009.

[19] J. Lehmann, and P. Hitzler. Foundations of refinement operators for description logics. In H. Blockeel and et al., editors, Proceedings of the 17th International Conference on Inductive Logic Programming, ILP2007, volume 4894 of LNCS, pages 161-174. Springer, 2008.

[20] J. Lehmann, and P. Hitzler. A refinement operator based learning algorithm for the ALC description logic. In H. Blockeel and et al., editors, Proceedings of the 17th International Conference on Inductive Logic Programming, ILP2007, volume 4894 of LNCS, pages 147-160. Springer, 2008.

[21] J. Lehmann, and P. Hitzler: Concept learning in description logics using refinement operators. Machine Learning 78(1-2): 203-250 (2010)

[22] J. Lehmann, L. Buehmann: ORE - A Tool for Repairing and Enriching Knowledge Bases. International Semantic Web Conference (2) 2010: 177-193

[23] J. Lehmann, S. Auer, L. Buehmann, S. Tramp: Class expression learning for ontology engineering. J. Web Sem. 9(1): 71-81 (2011)

Bibliography III

[24] J. Potoniec, A. Lawrynowicz: RMonto - towards KDD workflows for ontology-based data mining, Planning to Learn and Service-Oriented Knowledge Discovery Workshop at the ECML/PKDD-2011

[25] J. Potoniec, A. Lawrynowicz: RMonto: ontological extension to RapidMiner, Demo session at the International Semantic Web Conference 2011

[26] M. Sebag: Distance induction in first order logic. In S. Dzeroski and N. Lavrac, editors, Proceedings of the 7th International Workshop on Inductive Logic Programming, ILP1997, vol. 1297 of LNAI, pages 264-272, Springer, 1997.

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.

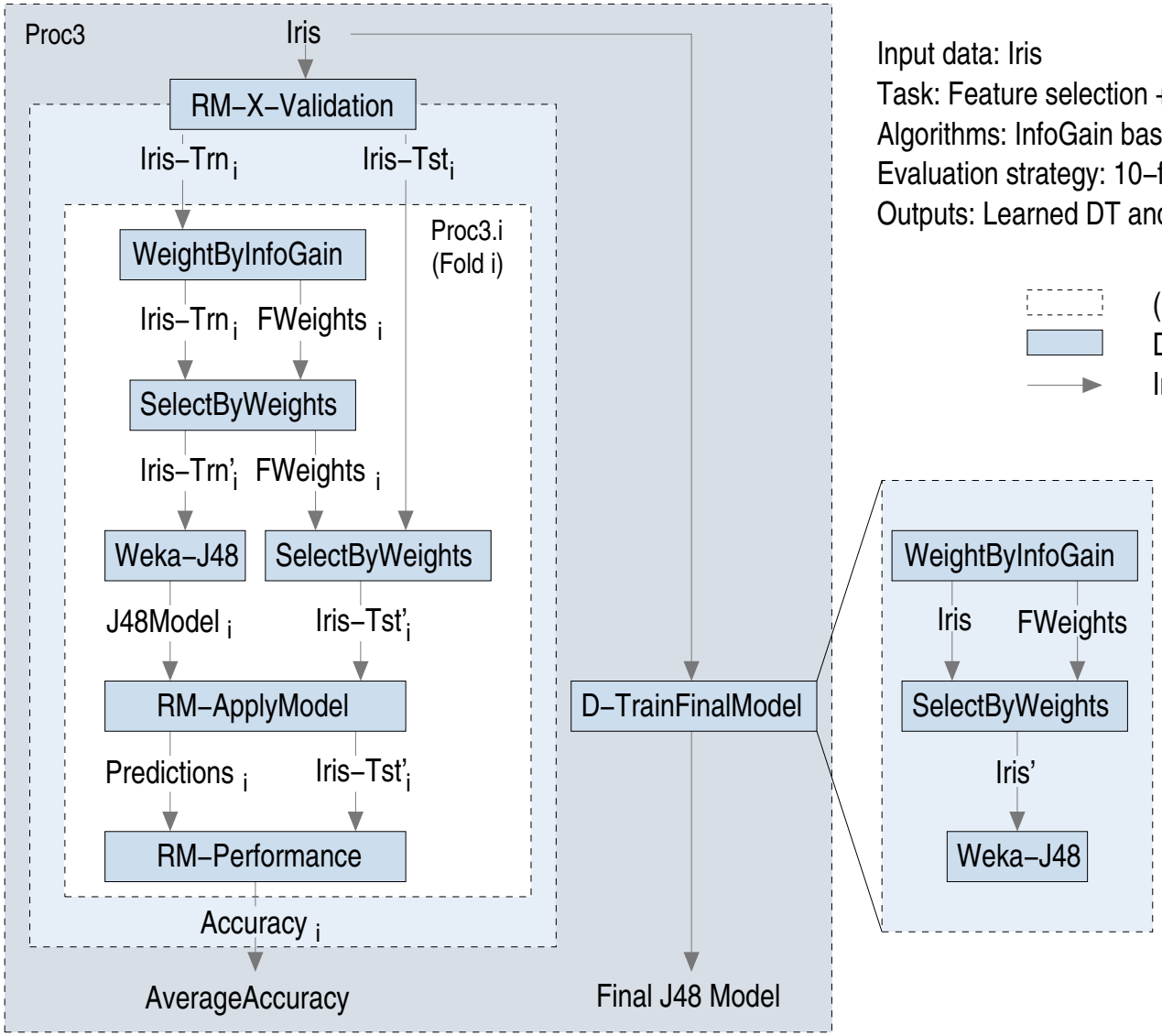
- **Greedy**: Restricted to the current (usually inductive) step of the DM process

- **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



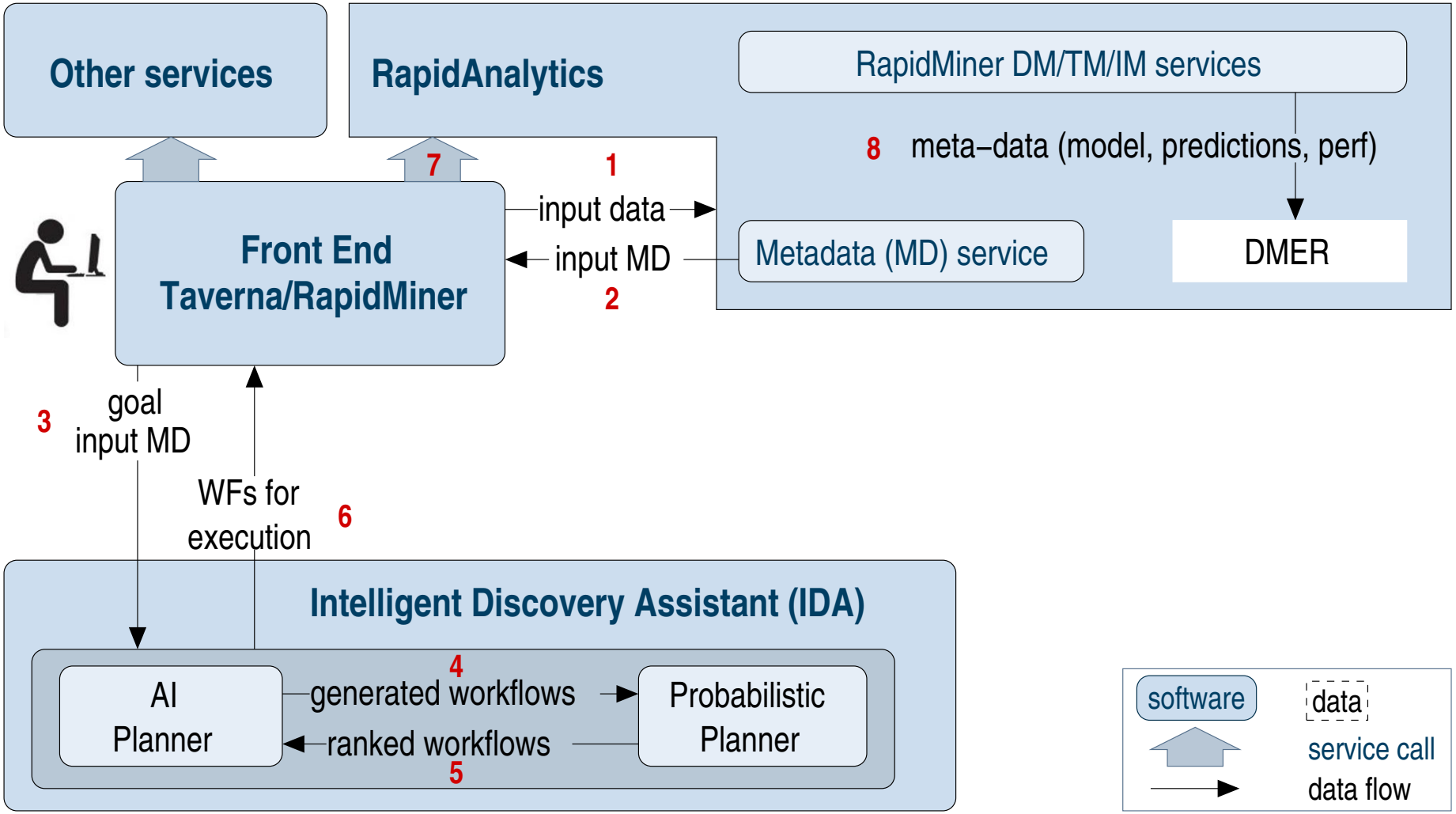
Input data: Iris
Task: Feature selection + classification
Algorithms: InfoGain based FS + DT
Evaluation strategy: 10−fold cross−val
Outputs: Learned DT and estimated accuracy

(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.

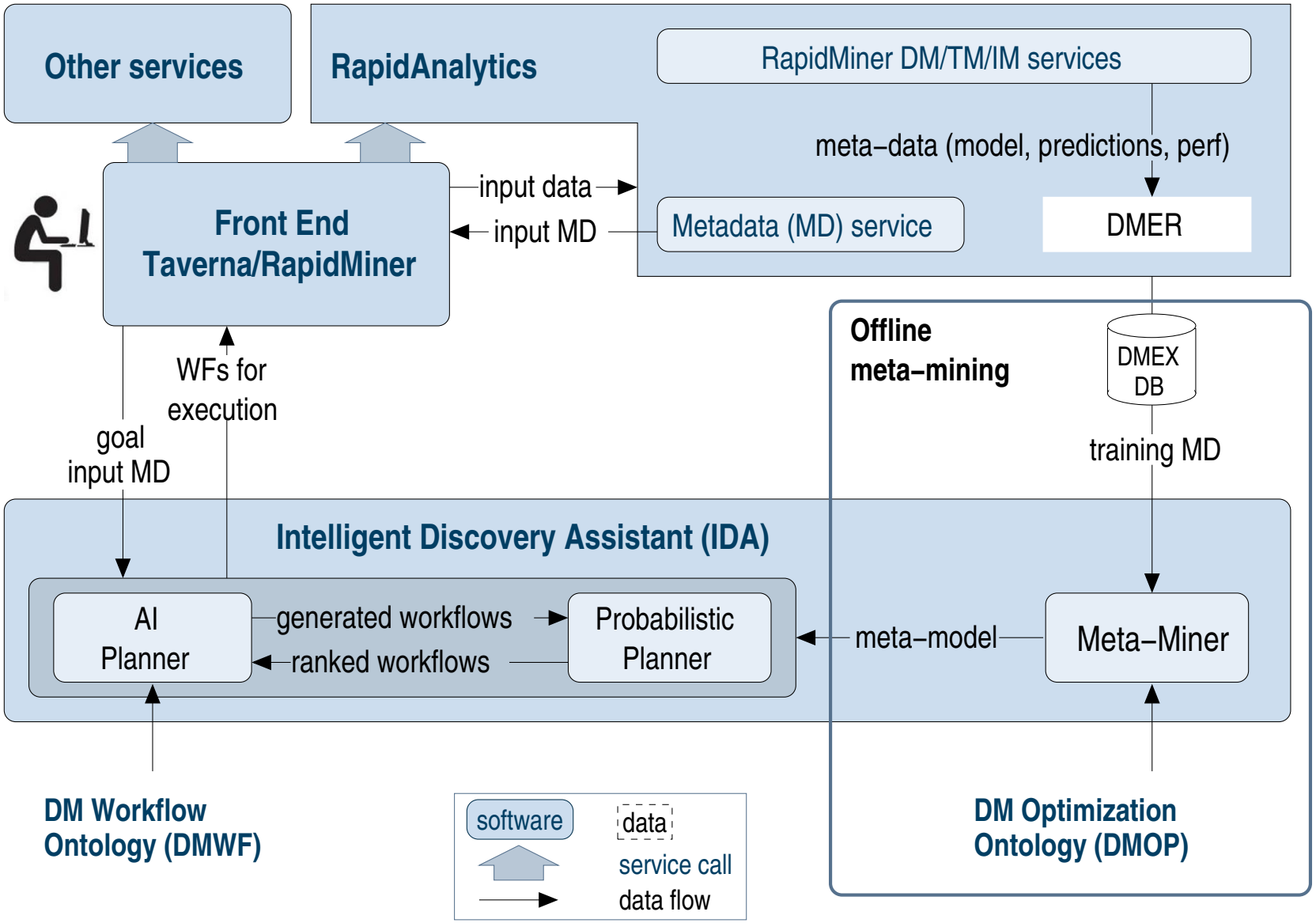
3-4. The IDA's basic AI Planner generates applicable workflows in a brute force fashion.

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

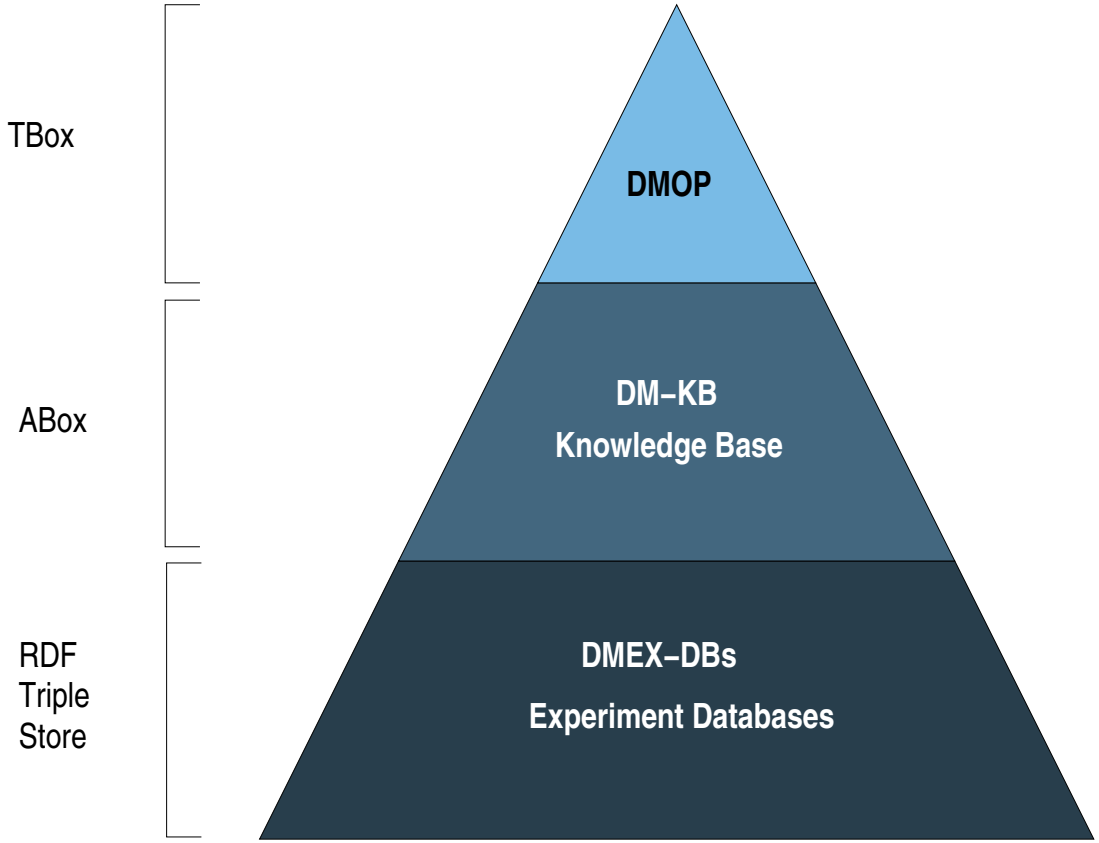
  $\Rightarrow$ higher-order feature space in which meta-learning can take place

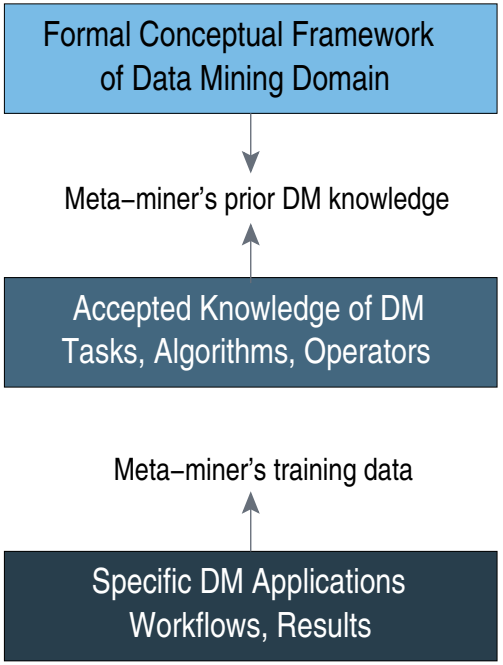
- **Approach**: model algorithms in terms of their underlying assumptions and other components of bias

  $\Rightarrow$ allows for generalization over algorithms and hence over workflows

  $\Rightarrow$ 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

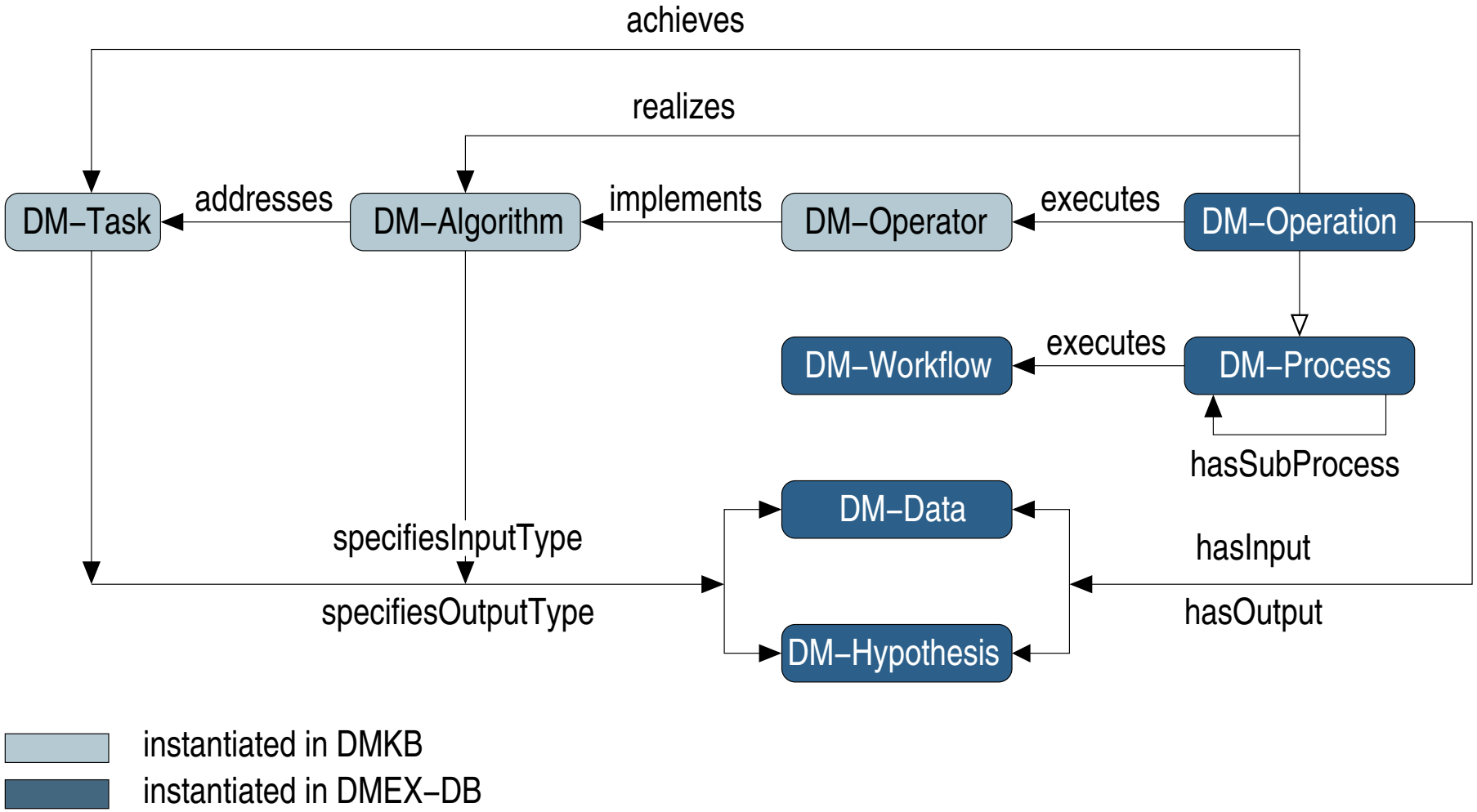
  $\Rightarrow$ illustrate our approach to breaking the algorithmic black box (p. 15)
  $\Rightarrow$ will serve as models for annotating new DM algorithm families

DM-KB (ABox)

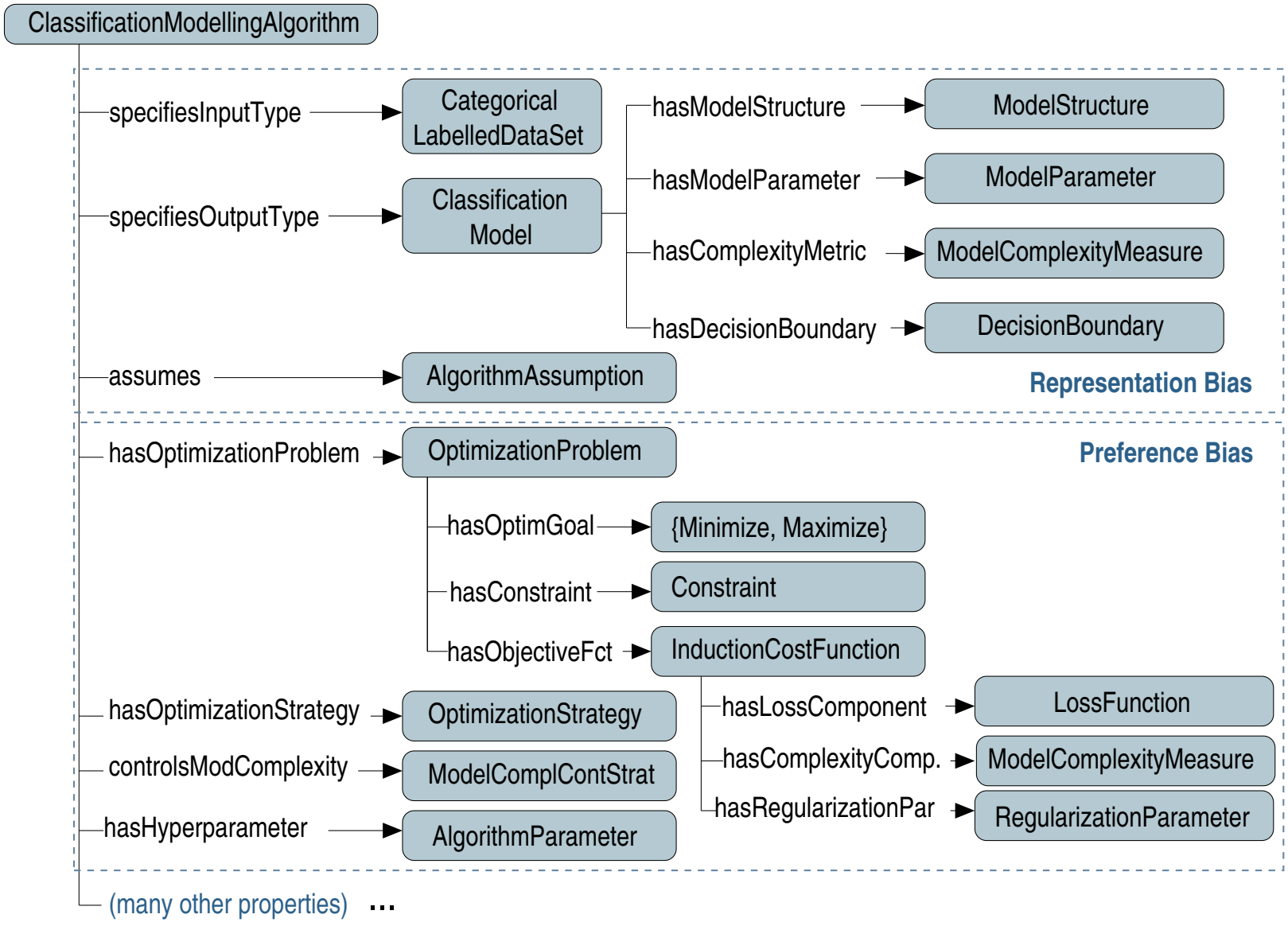
- describes individual algorithms using concepts from DMOP
- links available operators from known DM packages to their source algorithms

  $\Rightarrow$ generalized frequent pattern mining over WFs from DMER

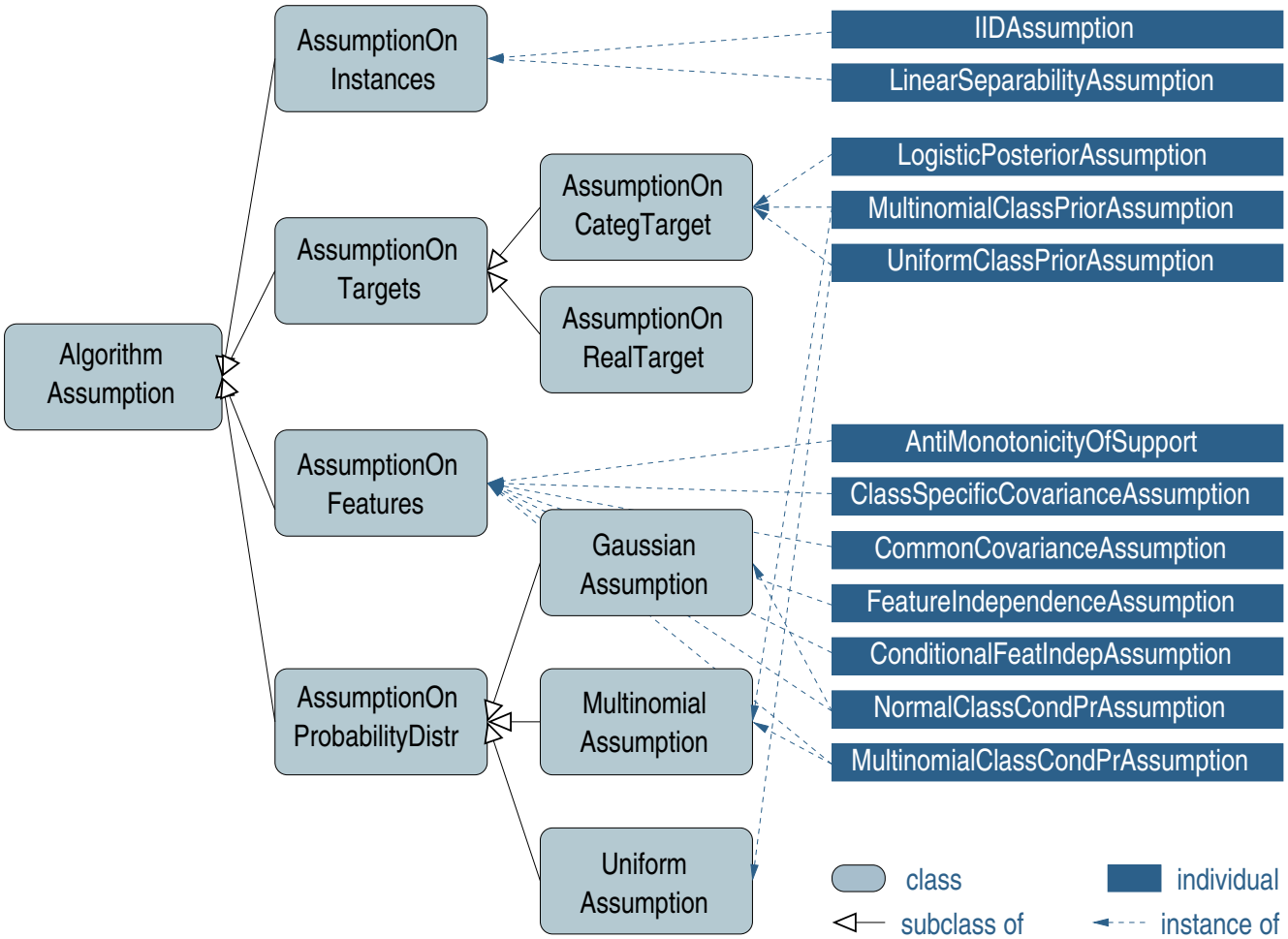
The Conceptual Framework



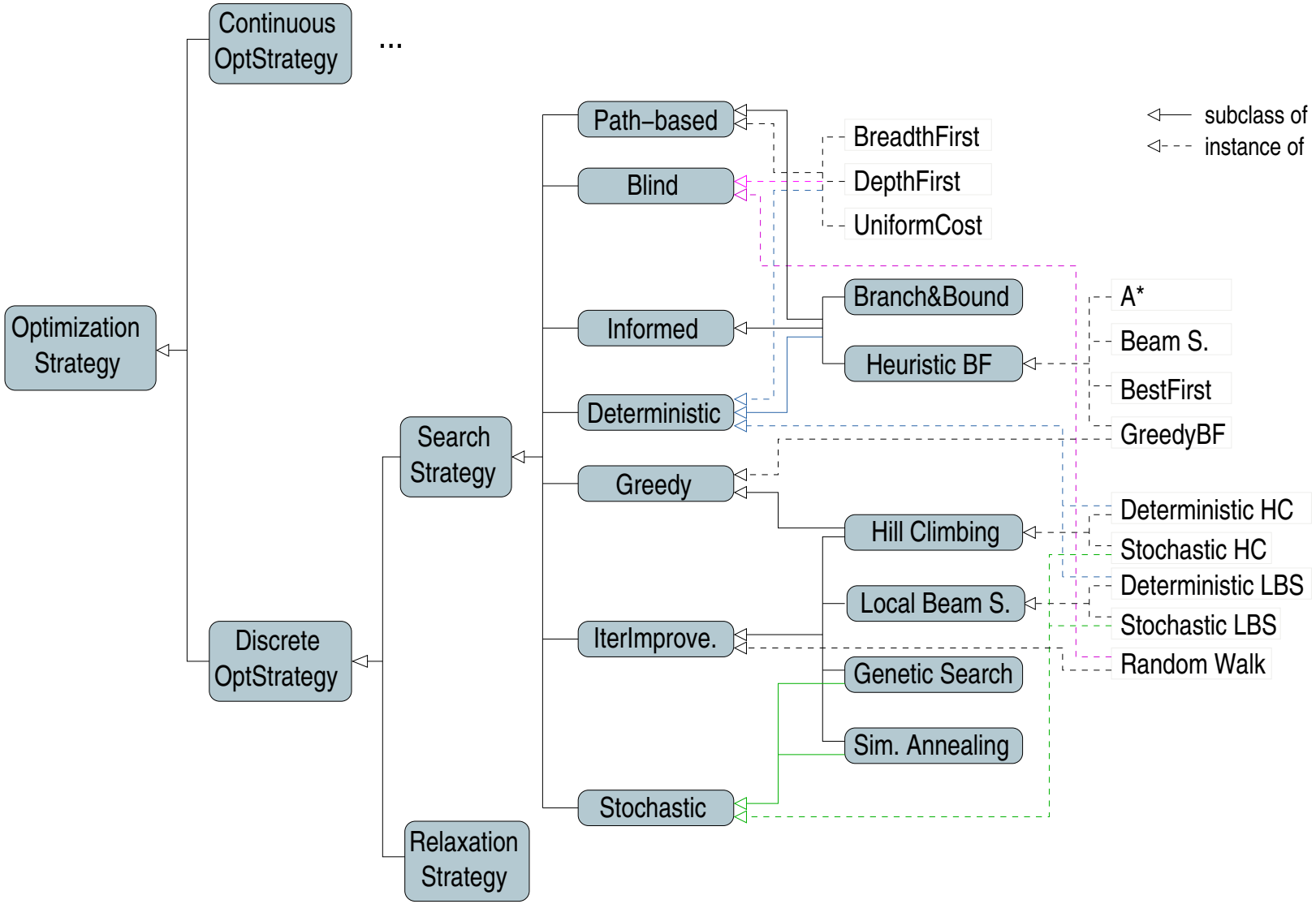
Inside Induction Algorithms



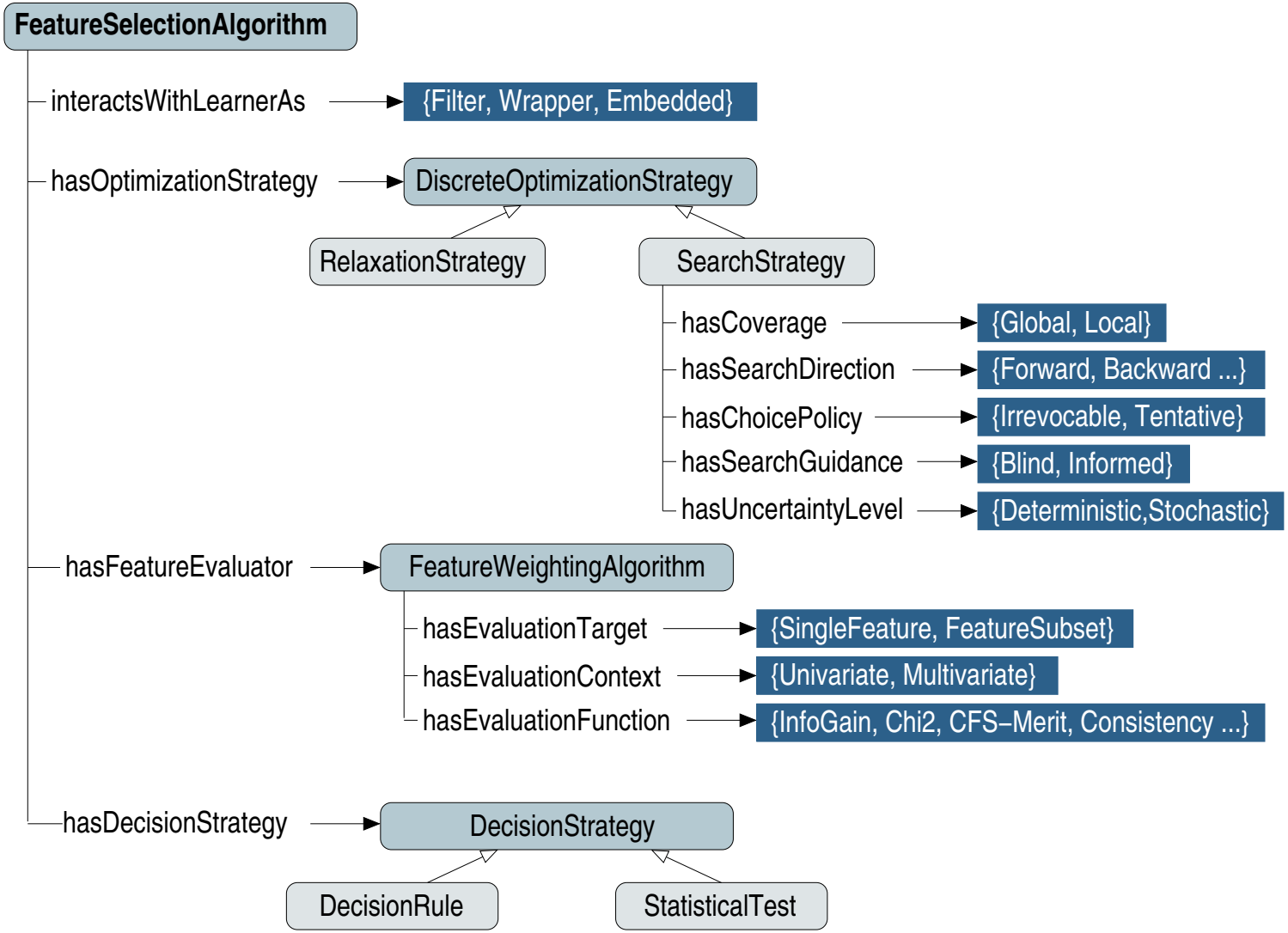
Algorithm Assumptions



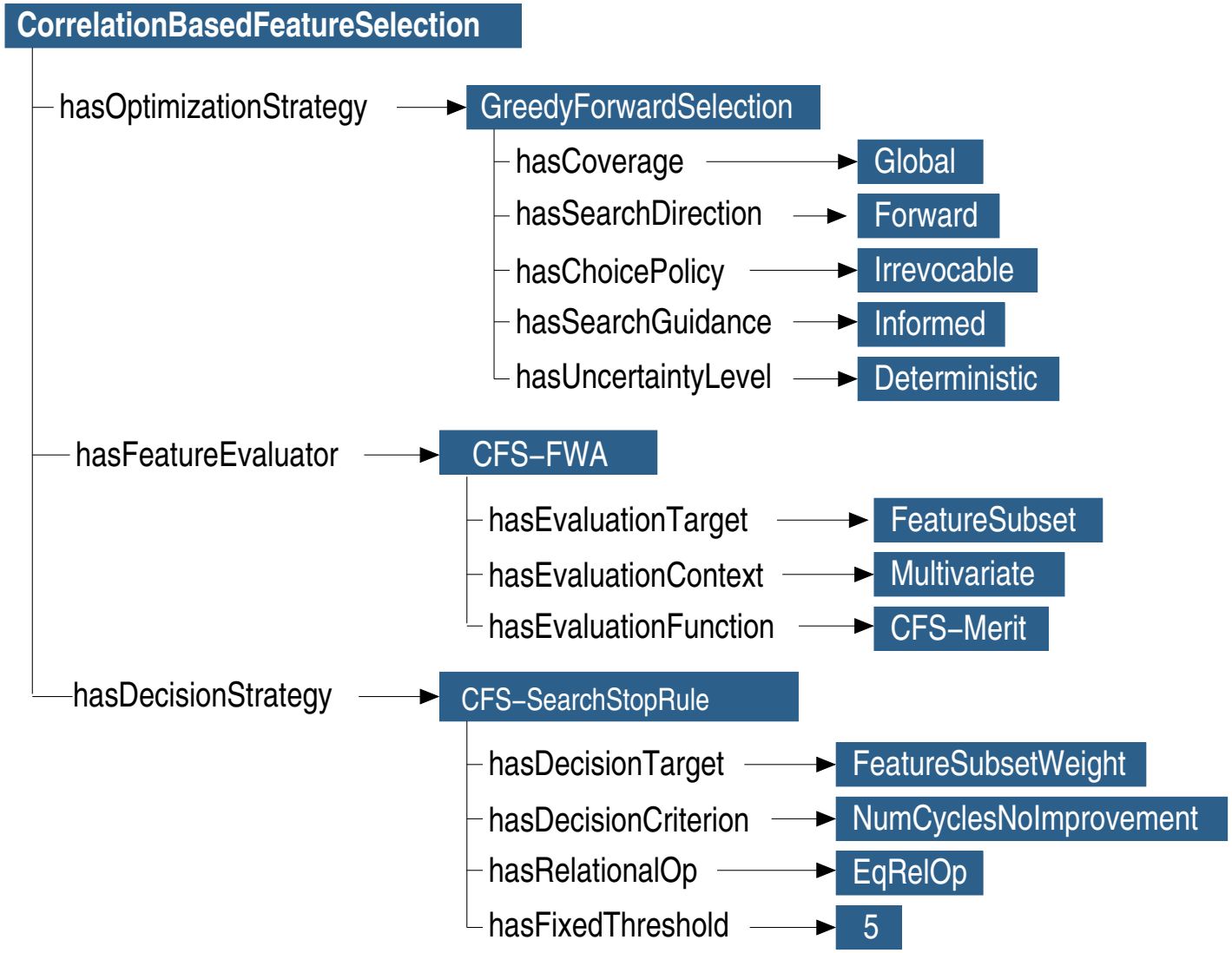
Optimization Strategies



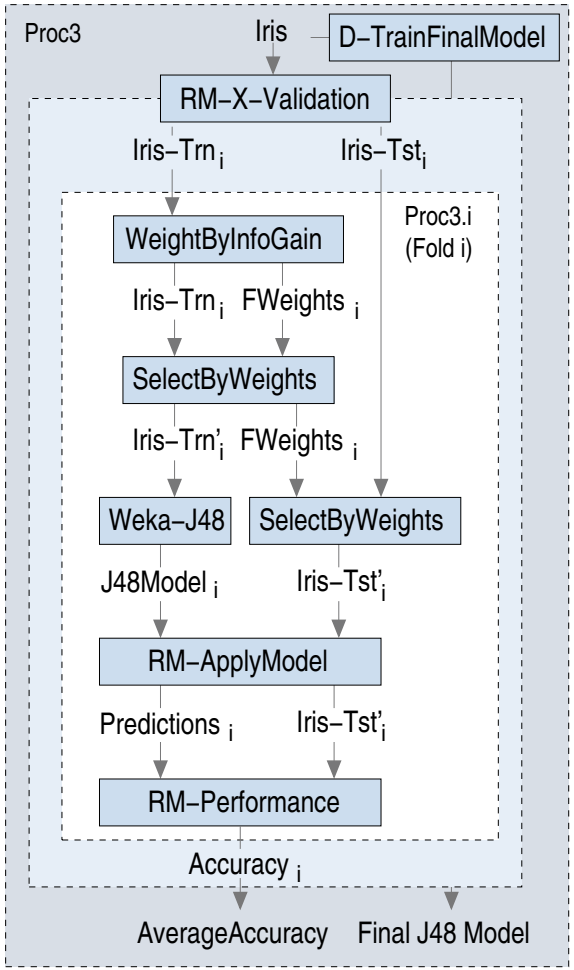
Feature Selection and Weighting



Example: Correlation-Based Feature Selection



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)
        ...
```

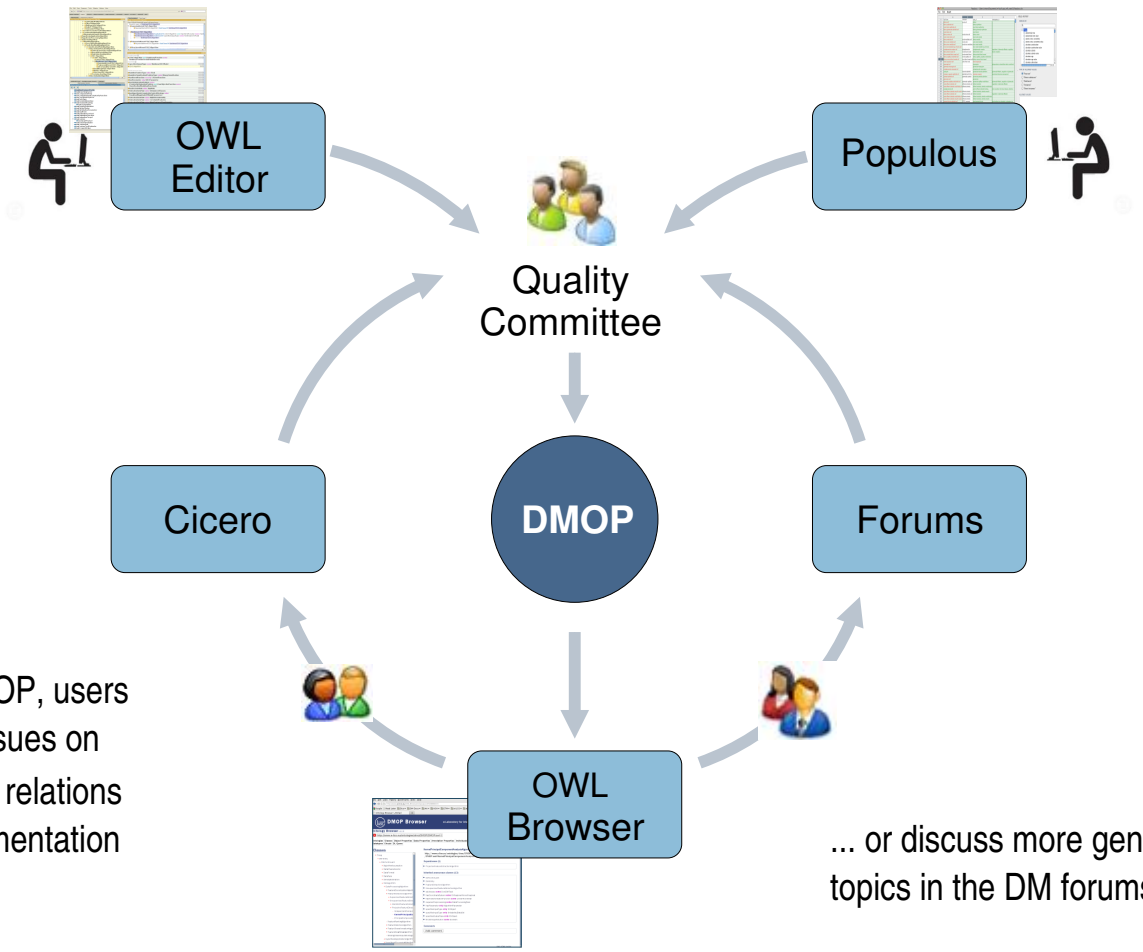
The DMOP CODeP

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

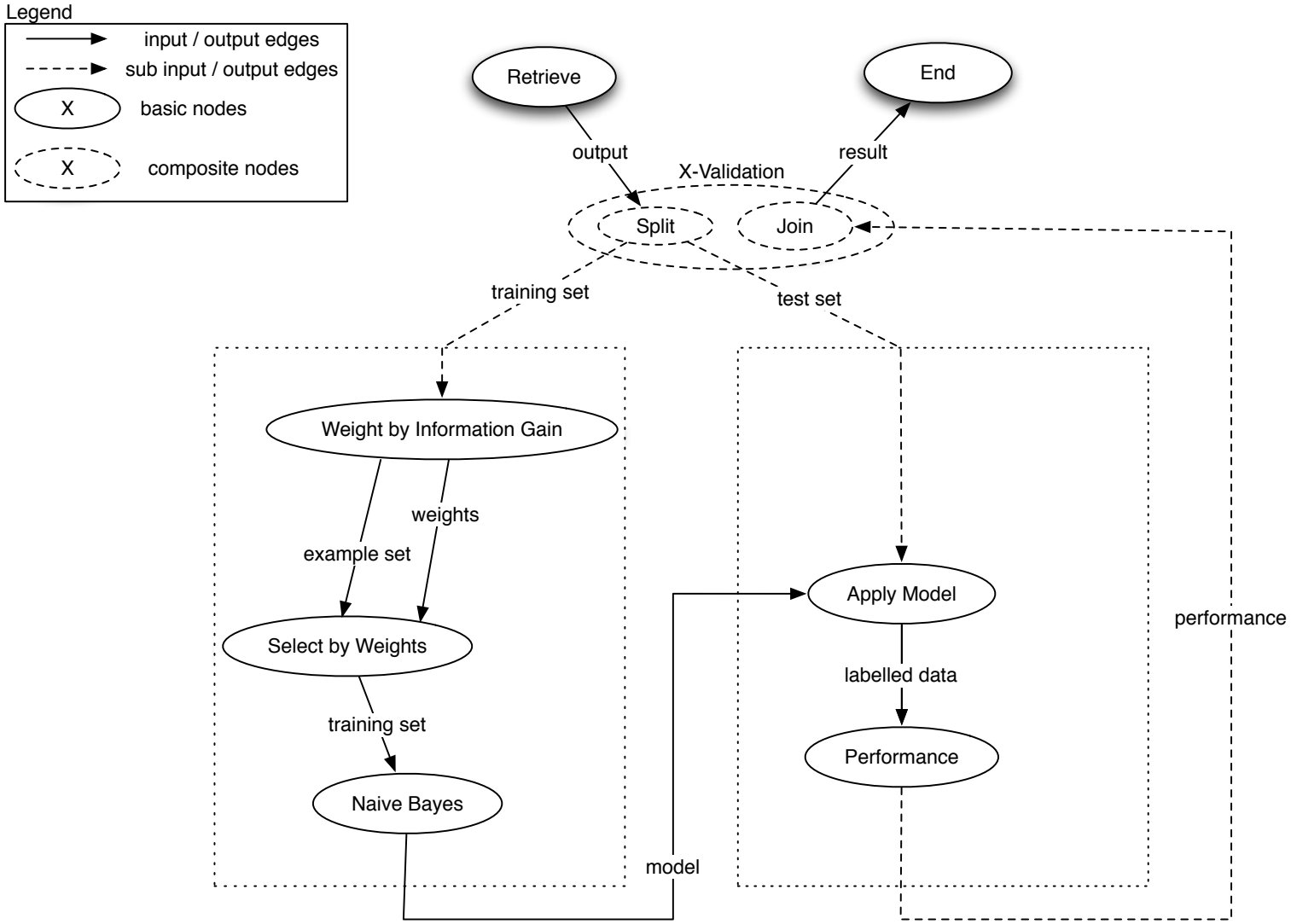
Data mining workflows representation

- 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.

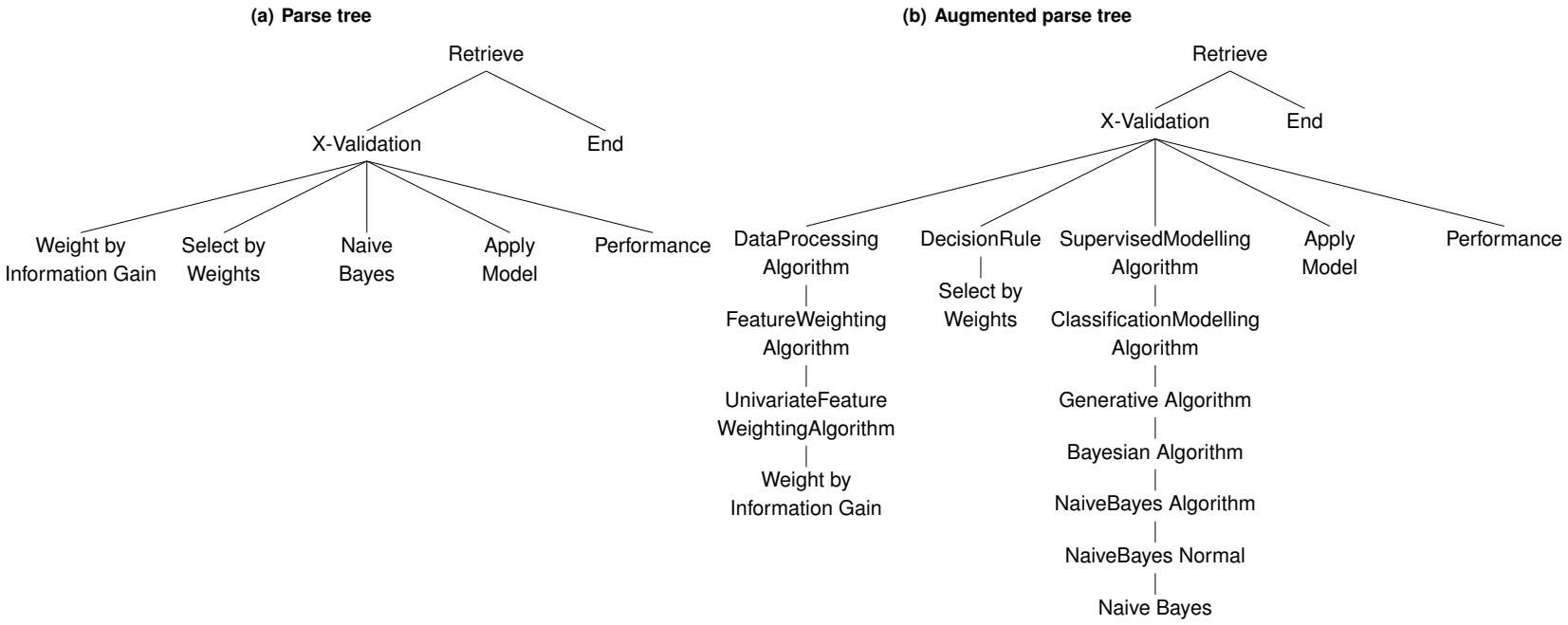
Frequent generalized pattern mining over workflows I

- 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



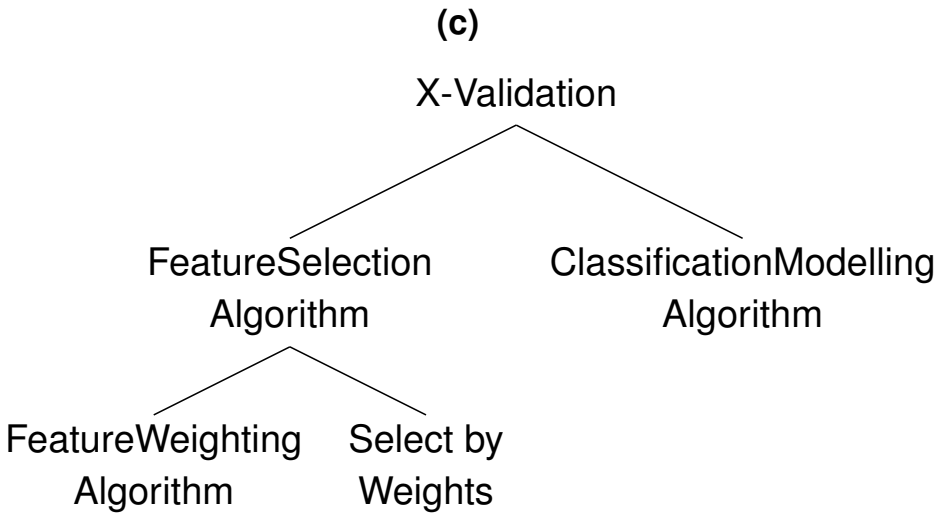
Parse and augmented parse tree of the previous WF

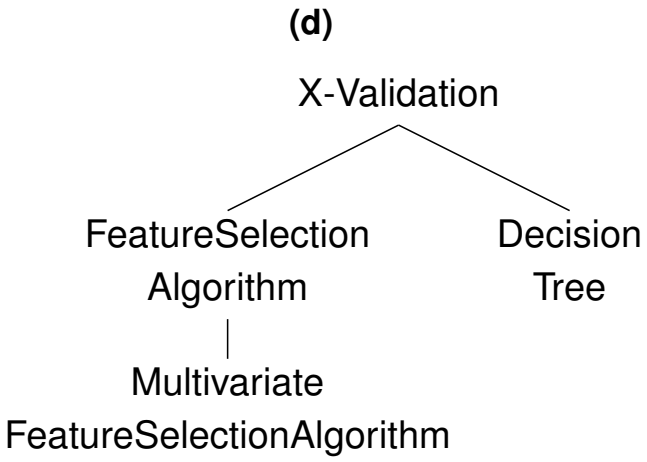


Generalized Frequent Pattern Extraction Results

- 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.