The Computer Journal Advance Access published June 4, 2012

doi:10.1093/comjnl/bxs057

Semantic Subgroup Discovery Systems and Workflows in the SDM-Toolkit

ANŽE VAVPETIČ[1,*] AND NADA LAVRAČ[1,2]

[1]*Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia*
[2]*University of Nova Gorica, Nova Gorica, Slovenia*
[*]*Corresponding author: anze.vavpetic@ijs.si*

This paper addresses semantic data mining, a new data mining paradigm in which ontologies are exploited in the process of data mining and knowledge discovery. This paradigm is introduced together with new semantic subgroup discovery systems SDM-search for enriched gene sets (SEGS) and SDM-Aleph. These systems are made publicly available in the new SDM-Toolkit for semantic data mining. The toolkit is implemented in the Orange4WS data mining platform that supports knowledge discovery workflow construction from local and distributed data mining services. On the basis of the experimental evaluation of semantic subgroup discovery systems on two publicly available biomedical datasets, the paper results in a thorough quantitative and qualitative evaluation of SDM-SEGS and SDM-Aleph and their comparison with SEGS, a system for enriched gene set discovery from microarray data.

Keywords: semantic data mining; relational data mining; inductive logic programming; domain knowledge; subgroup discovery; ontologies; microarray data

Received 25 November 2011; revised 12 March 2012
Handling editor: Einoshin Suzuki

1. INTRODUCTION

Knowledge discovery in databases (KDD) refers to the interactive and iterative process of finding interesting patterns and models in data [1]. The most common setting in knowledge discovery is rather simple: given is the empirical data and a data mining task to be solved, the data are pre-processed, then a data mining algorithm is applied and the end result is a predictive model or a set of descriptive patterns which can be visualized, interpreted and deployed in problem-solving tasks.

Data mining algorithms included in the contemporary data mining platforms such as Weka [2], KNIME [3], Orange [4] and RapidMiner [5] provide an extensive support for mining empirical data stored in a single table format, usually referred to as propositional data. These data mining platforms support all the most common propositional data mining tasks, including (but not limited to)

(i) classification and regression: predicting the value of the target attribute from the values of other attributes;
(ii) clustering: grouping objects into groups of similar objects;
(iii) association analysis: discovering correlations between sets of items which are most often found together in a set of transactions.

Data mining platforms like Weka provide their own implementations of the most popular and most commonly used data mining algorithms such as the C4.5 decision tree induction algorithm [6], the k-means clustering algorithm [7] and the Apriori association rule learning algorithm [8].

The task addressed in this paper is *subgroup discovery*, a data mining task at the intersection of classification and association discovery. The task of subgroup discovery was defined by Klösgen [9] and Wrobel [10] as follows: 'Given a population of individuals and a property of those individuals that we are interested in, find population subgroups that are statistically 'most interesting', for example, are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest'. Patterns discovered by subgroup discovery methods (called *subgroup descriptions*) are rules of the form $Class \leftarrow Conditions$, where the condition part of the rule is a logical conjunction of features (items, attribute values) or a conjunction of logical literals that are characteristic for a selected class of individuals or data objects.

It is well known from the literature on inductive logic programming (ILP) [11, 12] and relational data mining (RDM) [13] that the performance of data mining methods can be significantly improved if additional relations among the data objects are taken into account. In other words, the knowledge discovery process can significantly benefit from the domain (background) knowledge. A special form of background knowledge, which has not been exploited in the original ILP and RDM literature, is ontologies. Ontologies are consensually developed domain models that formally define the semantic descriptors and can act as a mean of providing additional information to machine learning (data mining) algorithms by attaching semantic descriptors to the data.

With the expansion of the semantic web and the availability of numerous ontologies, the amount of *semantic data* (data which include semantic information, e.g. ontologies and annotated data collections) is rapidly growing. Such domain knowledge is usually represented in a standard format that encourages knowledge reuse. Two popular formats are the web ontology language (OWL)[1] for ontologies and the resource description framework (RDF)[2] triplets for other structured data. This domain knowledge is usually consensual and built collaboratively by domain experts (e.g. by using Protégé,[3] a popular GUI tool for building ontologies).

The RDF data model is simple, yet powerful. A representation of the form *subject–predicate–object* ensures the flexibility of the data structures, and enables the integration of heterogeneous data sources. Data can be directly represented in RDF or (semi-)automatically translated from propositional representations to RDF as graph data. Consequently, more and more data from public relational data bases are now being translated into RDF as *linked data*.[4] In this way, data items from various databases can be easily linked and queried over multiple data repositories through the use of semantic descriptors provided by the supporting ontologies encoding the domain models and knowledge.

In data mining experiments, there are usually abundant empirical data available, but background knowledge is seldom used, since it usually cannot be directly employed. The data mining community is now faced with a new challenge of exploiting this vast resource of domain knowledge of semantically annotated data in the process of data mining and knowledge discovery. This paper uses the term *semantic data mining* to denote this new data mining challenge and approaches in which semantic data are mined.

Data mining methods can indeed be significantly improved by providing semantic descriptors to the data and by providing additional relations among data objects. By using ontologies, the induced hypotheses can be formed from terms that have been agreed upon by the domain experts. Moreover, in hypothesis construction, using higher level ontological concepts provides the means for more effective generalizations that would not have been possible by using only the terms used in instance descriptions. Semantic data mining has a great potential utility in many applications where ontologies are used as semantic descriptors for the data, for example, in biomedicine, biology, sociology, finance, where the number of available ontologies is rapidly growing.[5]

The algorithms implemented in the contemporary data mining platforms (e.g. Weka or Orange) currently focus on propositional data and the platforms do not support the inclusion of RDM and ILP algorithms which enable using background knowledge in hypothesis construction. The first step in this direction was done by incorporating the RSD algorithm [14] for relational subgroup discovery into the Orange4WS open-source data mining platform [15]. Orange4WS supports knowledge discovery workflow construction from distributed data mining services, enabling researchers and end-users to achieve the repeatability of experiments and simple sharing of workflows and system implementations. The work of this paper is a step toward enriching these data mining platform with a new functionality of semantic data mining, where domain ontologies are used as an additional information source for data mining.

In this paper, we present three approaches to semantic data mining. We first revisit a special purpose subgroup discovery system for analyzing gene expression microarray data, named SEGS (search for enriched gene sets) [16]. Next, we present two new domain-independent systems for semantic subgroup discovery, whose development was inspired by the success of SEGS:

(1) SDM-SEGS,[6] a domain-independent semantic subgroup discovery system based on SEGS,
(2) SDM-Aleph, a domain-independent semantic subgroup discovery system based on the ILP system Aleph.

These two systems implement numerous core components of the novel semantic data mining paradigm explained in this paper that builds on two previous papers [17, 18].

Compared with [17], this paper presents several improvements. The semantic subgroup discovery system g-SEGS (now named SDM-SEGS) is described in much more detail (the pseudo-code is provided as well), and we also present our new system SDM-Aleph. Both systems are now publicly available in a toolkit, named SDM-Toolkit, usable in the data mining platform Orange4WS [15]. We provide reusable workflows for an illustrative example and for two real-life use cases, showing the potential of our toolkit for practical knowledge discovery from microarray data. By comparing SEGS, SDM-SEGS and

[1] http://www.w3.org/OWL/.
[2] http://www.w3.org/RDF/.
[3] http://protege.stanford.edu/.
[4] See the Linked Data site http://linkeddata.org/.
[5] See http://bioportal.bioontology.org/.
[6] This system was named g-SEGS in our paper published in the Proceedings of Discovery Science Conference 2011 [17], and is here renamed for the elegance of unified systems naming.

SDM-Aleph on two biomedical domains, we provide a thorough quantitative and qualitative systems evaluation.

Like in the second paper upon which this paper is based [18], we use Orange4WS, here upgraded with SDM-SEGS and SDM-Aleph, which enables the use of ontologies in the data mining process. The advantage of using Orange4WS over other data mining toolkits like Weka, KNIME and RapidMiner is its service orientation and the availability of numerous data mining and data visualization algorithms enclosed in the original open source Orange data mining platform [4].

The main novelties of this paper are a refined definition of the task of semantic data mining, two new general purpose semantic subgroup discovery systems SDM-SEGS and SDM-Aleph, and a first semantic data mining toolkit, named SDM-Toolkit, which has been made publicly available. Other contributions of this paper are as follows. We have revisited a successful domain-specific system SEGS in the context of semantic data mining. The use of SDM-Toolkit tools for biomedical workflow construction and their execution in the service-oriented data mining environment Orange4WS is show-cased on an illustrative example and two biomedical real-life problem domains. We also provide a qualitative evaluation of the SDM-SEGS and SDM-Aleph systems, supported by experimental results and comparisons with SEGS. The contribution of this paper is the insight that SEGS and SDM-SEGS are more appropriate for data analysis in biological and biomedical domains where rule specificity is desired, while SDM-Aleph is a more general purpose system, resulting in more general rules of higher precision.

Despite the fact that SDM-SEGS and SDM-Aleph are not limited to applications in biology, two such real-life domains were used in our experiments to assess the characteristics of the systems in comparison with the baseline system SEGS whose application is limited to biology (microarray data analysis).

The paper is organized as follows. In Section 2, we present a refined definition of the task of semantic data mining, together with three semantic subgroup discovery systems: SEGS, SDM-SEGS and SDM-Aleph. Section 3 provides an illustrative example of using these systems in the data mining platform Orange4WS. Section 4 presents two biomedical domains, acute lymphoblastic leukemia (ALL) and human mesenchymal stem cells (hMSC), together with the developed biomedical workflows and a detailed quantitative and qualitative comparison of the three systems. Section 5 presents the related work. The paper concludes with a discussion and directions for further work.

2. SEMANTIC DATA MINING

In this section, we define the semantic data mining task, describe an existing system SEGS, followed by the descriptions of two new semantic subgroup discovery systems SDM-SEGS and SDM-Aleph.

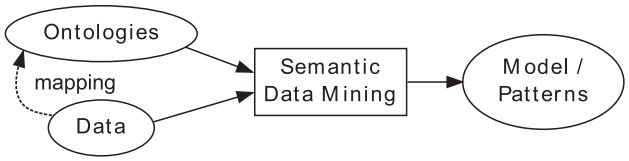


FIGURE 1. Schema of the semantic data mining task, with ontologies and annotated data as inputs.

2.1. Semantic data mining task definition

The term *semantic subgroup discovery* was first introduced in [19] and was later extended to semantic data mining in [17]. Semantic data mining can be defined as follows:

Given: a set of domain ontologies, and empirical data annotated by domain ontology terms,

Find: a hypothesis (a predictive model or a set of descriptive patterns), expressed by domain ontology terms, explaining the given empirical data.

Liu [20] has proposed his own definition of semantic data mining: 'We propose to exploit the advances of the semantic web technologies to formally represent domain knowledge including structured collection of prior information, inference rules, knowledge enriched datasets etc, and thus develop frameworks for systematic incorporation of domain knowledge in an intelligent data mining environment. We call this technology the semantic data mining'. His definition is too broad to be used for the needs of this paper. Consequently, we propose a more refined definition of semantic data mining below.

Given: *domain knowledge* in the form of ontologies, a set of *training examples* (experimental data), and an *example-to-ontology mapping* which associates each example with appropriate ontological concepts.

Find: a hypothesis (a predictive model or a set of descriptive patterns), expressed by domain ontology terms, explaining the given empirical data.

In the following subsection, each of the systems is described by instantiating this general framework. The task of semantic data mining is illustrated in Fig. 1.

2.2. Existing SDM system SEGS

A domain-specific system that uses ontologies and other hierarchies as background knowledge for data mining is SEGS, which upgrades previous approaches to gene set enrichment analysis [21, 22]. Compared with earlier work in gene set enrichment[7] [21, 22], the novelty of SEGS is that it does not only

[7] A gene set is *enriched* if the genes that are members of this gene set are statistically significantly differentially expressed compared with the rest of the genes.

test existing gene sets (existing ontology terms) for differential expression but it generates also new gene set descriptions as conjunctions of ontological concepts that may represent novel biological hypotheses.

SEGS can be described in terms of the SDM framework from Section 2.1 as follows:

(1) *domain knowledge* is an internal representation of the Gene Ontology[8] (GO) and Kyoto Encyclopedia of Genes and Genomes[9] (KEGG);
(2) *training data* is a list of ranked genes;
(3) *example-to-ontology mapping* associates each gene with a number of GO and KEGG concepts;
(4) additionally, a binary relation *interacts* is used, which models gene–gene interactions.

The basic rule construction idea of SEGS is the same as the one used in the new general purpose system SDM-SEGS (described in the next section). The resulting rules are statistically evaluated using three measures relevant for biological domains: the Fisher's exact test [23], PAGE [22] and GSEA [21].

The drawback of SEGS in terms of semantic data mining is that it is domain specific due to the fact that the ontologies and interaction data used are fixed to the GO and KEGG, stored in a native format. SDM-SEGS presented in the following section does not have these limitations. Note that, on the other hand, the domain specificity enables SEGS to be better tuned to the specific task of analyzing microarray data.

2.3. SDM-SEGS

This section describes our new semantic subgroup discovery system SDM-SEGS that can be used to discover subgroup descriptions from ranked data as well as from general class-labeled data with the use of input OWL ontologies. The ontologies are exploited in a manner similar as in SEGS (i.e. ontological concepts are used as terms that form rule conjuncts), with the important difference that they can be (a) from any domain and (b) in a standard OWL format. However, it uses at most four input ontologies and the user can specify only one additional relation between the examples, due to the limitations imposed by the original SEGS algorithm.

Below we describe the main parts of SDM-SEGS: the input data, the hypothesis language, the rule construction algorithm, the rule selection and evaluation principles and its implementation.

2.3.1. *Input*

Apart from various parameters (e.g. for controlling the minimum support criterion, the maximum rule length, etc.), the main inputs are

(1) *domain knowledge* in the legacy SEGS format or in the form of OWL ontologies;[10]
(2) *training data* which is a list of class-labeled or ranked examples;
(3) *example-to-ontology mapping* which associates each example with a number of concepts from the ontologies and
(4) binary relation *interacts*, which is a list of pairs of identifiers of examples which interact in some way.

In the case of class-labeled data, the user specifies the target class and in the case of ranked examples, the user specifies a threshold value, which splits the examples into two classes (positive and negative) according to their rank. In both cases, we can treat the problem as a two-class problem.

The example-to-ontology mapping is used to associate each input example with the ontological concepts that the example is annotated with.

2.3.2. *Hypothesis language*

The hypothesis language is a set of rules of the form $class(X) \leftarrow Conditions$, where $Conditions$ is a logical conjunction of terms that represent ontological concepts.

As an illustration, a possible rule can have the following form

$$class(X) \leftarrow doctor(X) \wedge germany(X).$$

Both $doctor$ and $germany$ are terms that represent ontological concepts $doctor$ and $germany$. If the input examples are people, this rule describes a subgroup of people who are doctors and live in Germany.

2.3.3. *Rule construction*

A set of rules that satisfy the size constraints (minimum support and maximum number of rule terms) is constructed using a top-down bounded *exhaustive* search algorithm shown in Fig. 2, which enumerates all the possible rules by taking one term from each ontology. The rule construction procedure starts with a default rule $class(X) \leftarrow$, which covers all the examples. Next, the algorithm tries to conjunctively add the top concept of the first ontology and if the new rule satisfies all the size constraints, it adds it to the rule set and recursively tries to add the top concept of the next ontology. In the next step, all the child concepts of the current term/concept are tried by recursively calling the procedure. Due to the properties of the *subClassOf* relation between concepts in the ontologies, the algorithm can employ an efficient pruning strategy. If the currently evaluated rule does not satisfy the size constraints, the algorithm can prune all the rules that would have been generated if this rule were further specialized.

Further gains can be achieved by skipping concepts that the user deems to be too general to be useful. These concepts can

[8] http://www.geneontology.org/.
[9] http://www.genome.jp/kegg/.

[10] Unlike SDM-Aleph described in Section 2.4, SDM-SEGS exploits only the *concept* and *subClassOf* relations.

```
function construct(rule, conj, k):
 # rule - the rule to specialize.
 # conj - the concept to add to the rule.
 # k - 'conj' is from the k-th ontology.

 # The set described by the current rule.
 newSet = intersect(set(rule), set(conj))

 # Is the set big enough?
 if newSet.size > MIN_SIZE:
   rule.add(conj)
   if clean(rule).size < MAX_TERMS and
      clean(rule).size > 0:
     rules.add(rule)

   # Can the rule be extended?
   if rule.size < max(MAX_TERMS, MAX_ONT):
     construct(rule, ontologies[k+1], k+1)
     rule.remove(conj)

     # Extend the rule with all successors.
     for each child in children(conj):
       if set(child).size > MIN_SIZE:
         construct(rule, child, k)

 # Also check the interacting set.
 interactingSet =
  intersect(set(rule), interacts(set(conj)))
 if interactingSet.size > MIN_SIZE:
   rule.add('interacts(' conj ')')
   if clean(rule).size < MAX_TERMS:
     rules.add(clean(rule))

 return rules
```

FIGURE 2. Rule construction procedure of SDM-SEGS.

be specified either by listing them directly or by specifying the level in the *subClassOf* hierarchy up to which the concepts are too general.

Additionally, the user can specify another relation between the input examples: the *interacts* relation. Two examples are in this relation, if they interact in some way (if the examples are people, we can say, for example, that two people are in the *interacts* relation if they are married). For each concept, which the algorithm tries to conjunctively add to the rule, it also tries to add its interacting counterpart. For example, if the current rule is $class(X) \leftarrow c_1(X)$ and the algorithm tries to add the term/concept $c_2(X)$, then it also tries to append the terms $interacts(X, Y) \wedge c_2(Y)$. For example, the antecedent of the rule

$$class(X) \leftarrow c_1(X) \wedge interacts(X, Y) \wedge c_2(Y)$$

```
function ruleSelection(examples, k):
 # examples - example set.
 # k - an example can be covered max k times.

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

 # Re-compute the WRAcc, ignore the weights.
 for each rule in resultSet:
   rule.score = WRAcc(rule)

 return resultSet
```

FIGURE 3. Rule selection procedure of SDM-SEGS.

can be interpreted as: all the examples which are annotated by concept c_1 and interact with examples annotated by concept c_2.

If we return to our example, where *interacts* could be interpreted as two people being married, then another example could be

$$class(X) \leftarrow interacts(X, Y) \wedge doctor(Y),$$

which describes all the persons which are married to a doctor.

2.3.4. *Rule selection*

As the number of generated rules can be large, uninteresting and overlapping rules have to be filtered out. In SDM-SEGS, rule selection is performed during rule post-processing using a weighted covering algorithm that selects the best rules according to the wWRAcc (weighted relative accuracy with example weights) heuristic [24]. The weighted covering algorithm uses example weights as means for considering different parts of the example space when selecting the best rules. The weighted covering algorithm used for rule selection is presented in Fig. 3, followed by the formula for computing the wWRAcc heuristic.

The wWRAcc heuristic is based on WRAcc, the heuristic known from CN2-SD subgroup discovery [24], which trades-off rule coverage and precision. The WRAcc heuristic is defined as

$$\mathrm{WRAcc}(C \leftarrow \mathrm{Cnd}) = \frac{n(\mathrm{Cnd})}{N} \cdot \left(\frac{n(\mathrm{Cnd} \wedge C)}{n(\mathrm{Cnd})} - \frac{n(C)}{N} \right),$$

where N is the number of all examples, $n(C)$ is the number of examples of class C, $n(\mathrm{Cnd})$ is the number of all covered

examples and $n(\text{Cnd} \wedge C)$ is the number of all correctly covered examples of class C.

The wWRAcc heuristic (defined below) adapts WRAcc to take example weights into account. It is defined as

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

where N' denotes the sum of weights of all examples, $n'(C)$ is the sum of weights of examples of class C, $n'(\text{Cnd})$ is the sum of weights of all covered examples and $n'(\text{Cnd} \wedge C)$ is the sum of weights of all correctly covered examples of class C.

Rule selection proceeds as follows. It starts with a set of generated rules, a set of examples with weights equal to 1 and parameter k, which denotes how many times an example can be covered before being removed from the example set. In each iteration, we select the rule with the highest wWRAcc value, add it to the final rule set and remove it from the set of generated rules. Then the counter m is increased to $m + 1$ and weights of examples covered by this rule decreased to $1/(m + 1)$, where example weight $1/m$ means that the example has already been covered by $m < k$ rules. These steps are repeated until the algorithm runs out of examples or rules or if no rule has a score above zero. Once the learning process is finished and the rules have been generated and filtered, they are evaluated using the original WRAcc measure.

2.3.5. *Implementation*

SDM-SEGS is written in C (the rule construction and selection parts) and Python (the user interface and web-service related code). It is implemented as a web service with an easy-to-use user interface in the Orange4WS service-oriented data mining platform, which upgrades the freely available Orange data mining environment. Orange4WS offers a large collection of data mining and machine learning algorithms and powerful visualization components. Additional components can be easily added by implementing them in Python or C/C++ or by directly importing an existing web service. All these components (*widgets*) can then be combined into workflows to solve a desired task.

Such an implementation enables the repeatability of experiments and simplifies the sharing of workflows and implementations. We provide an illustrative example workflow using SDM-SEGS in Section 3.2 and a real-life biomedical workflow in Section 4.2.

2.4. SDM-Aleph

In this section, we present our new semantic subgroup discovery system SDM-Aleph, based on the ILP system Aleph.[11] SDM-Aleph was designed to be used in a similar way as SDM-SEGS. SDM-Aleph can discover subgroup descriptions for class labeled or ranked data with the use of input OWL ontologies as domain knowledge, where the ontological concepts are used as rule conjuncts. Unlike SDM-SEGS which only takes four ontologies as input and only one additional *interacts* relationship, in SDM-Aleph any number of ontologies and additional relations between the input examples can be specified, which is due to the powerful underlying first-order logic formalism of the ILP system Aleph.

In the following paragraphs, we describe the input to our system, its hypothesis language, the used rule construction and selection techniques and its implementation details.

2.4.1. *Input*

The required inputs to the system are similar to the ones in SDM-SEGS, but less constrained:

(1) *domain knowledge* in the legacy SEGS format or in the form of OWL ontologies (where the *concept* and *subClassOf* relations are used, as well as other binary relations between ontology terms, which hold for all members of the ontology concepts[12]);
(2) *training data* which is a list of class-labeled or ranked examples;
(3) *example-to-ontology mapping* which associates each example with a number of concepts from the ontologies and
(4) optionally, *additional binary relations* between input examples, specified extensionally as pairs of example identifiers.

2.4.2. *Hypothesis language*

The hypothesis language is also similar to the one of SDM-SEGS. The hypothesis language is again a set of rules of the form $class(X) \leftarrow Conditions$, where $Conditions$ is a logical conjunction of unary and binary predicates. The unary predicates represent ontological concepts, while the binary predicates represent binary relations between some of the input examples. The user can add any number of additional binary relations to the hypothesis language, but by doing so the hypothesis search space will significantly increase. Note that with SDM-Aleph, the user can specify not only the $interacts$ relation, but an arbitrary number of relations between the examples.

2.4.3. *Rule construction and selection*

The basic rule construction method follows the original Aleph implementation. Through specific settings, we have tailored the search procedure to the context of semantic subgroup discovery.

The main four steps are the following, summarized based on the Aleph manual:[13]

(1) *Select example.* Select one of the examples.

[11] http://www.cs.ox.ac.uk/activities/machinelearning/Aleph/.

[12] Binary relations which hold for all members of two ontology concepts can be added to the background knowledge intensionally as a Prolog binary predicate definition.

[13] http://www.cs.ox.ac.uk/activities/machinelearning/Aleph/.

(2) *Build the most specific clause.* Construct the most specific clause that logically entails the selected seed example, and is within the provided language constraints (the maximum rule length)—this clause is usually called a *bottom clause*. More details regarding the construction of a bottom clause can be found in [25].
(3) *Search.* Find a clause more general than the bottom clause. This step enumerates the acceptable clauses within the given constraints (minimum support) by using a best-first strategy using a heuristic function selected by the user.
(4) *Remove redundant.* The clause with the best score found in the previous step is added to the final rule set (a model).

As mentioned before, Aleph provides settings which can affect each of the four steps through various parameters. In order to get a model satisfactory to our task at hand, we limit the maximum rule length and the minimum support of a rule to the user's preference, we handle noise by allowing imperfect rules to avoid model over-fitting and for the *search* step we use heuristic search guided by the WRAcc heuristic. Regarding the *remove redundant* step, we use the `induce_cover` mode, where the procedure removes examples covered by the best clause only from the set of possible seeds for constructing bottom clauses. The consequence of this is that the resulting rules may overlap in terms of covered examples, which is common in subgroup discovery.

2.4.4. *Implementation*

SDM-Aleph is written in Prolog (the original Aleph code) and in Python (the user interface, web-service related code and the SDM-related code). It is implemented as a web service with an easy-to-use user interface in Orange4WS. SDM-Aleph can be used in workflows interchangeably with SDM-SEGS. The benefits of such an implementation are, of course, identical as in the case of SDM-SEGS.

SDM-Aleph involves multiple layers of processing. First, the inputs (ontologies, examples and the example-to-ontology mapping) need to be converted to a proper Horn clause form. Here, we present the main ideas.

Each ontological concept `c`, with child concepts `c1, c2, .., cm`, is encoded as a unary predicate `c/1`:

```
c(X) :- c1(X) ; c2(X) ; ... ; cm(X).
```
[14]

Each child concept is defined in the same way. To encode the whole ontology, we need to start this procedure at the root concept. All these predicates are allowed to be used in the rule body and are tabled for faster execution.

Each example is encoded as an atom defined for the concepts with which it is annotated. If the kth example is annotated by concepts `c1,c2,...,cm` (this is defined by the example-to-ontology mapping), we encode it as a set of ground facts:

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

Any binary relation r between input examples is modeled by adding the `r/2` predicate to the hypothesis language and defining it extensionally.

3. SDM WORKFLOWS IN ORANGE4WS

In this section, we present an illustrative problem domain and demonstrate typical usage of the developed semantic data mining tool in the Orange4WS platform. We also provide a link to this publicly available SDM-Toolkit.

3.1. Illustrative example

As a proof-of-concept semantic data mining example [17], consider a bank which has the following data about its customers: place of living, employment, bank services used, which includes the account type, possible credits and insurance policies and so on. The bank also categorized the clients as 'big spenders' or not and wants to find patterns describing big spenders. Table 1 presents the training data. Suppose we also have three ontologies: an ontology of banking services, an ontology of locations and an ontology of occupations, shown in Fig. 4.

We wish to use these ontologies as domain knowledge in the process of subgroup discovery in the given dataset. In order to do so, we need a mapping between the input examples and concepts in the domain ontologies. In this illustrative use case, each value from the dataset corresponds to one concept from the ontologies, e.g. if we have an example with attribute value `occupation='Doctor'`, then we annotate this example with ontological concept `Doctor`. Using this information, the learning algorithm can further generalize the data using more general ontological concepts. For instance, because the previously mentioned person is a `Doctor`, then according to the occupation ontology he also works in the `Health` sector.

An important fact here is that an algorithm can, using this domain knowledge, construct subgroup descriptions from concepts which are more general and do not appear in the data itself. A possible pattern in this domain could be, e.g. $big\_spender(X) \leftarrow germany(X)$, describing all examples/people living in Germany, although in the data table we only have the information only on specific German cities.

Figure 5 presents a subset of subgroup descriptions discovered on the banking domain.

3.2. Workflow construction in Orange4WS

In this section, we demonstrate how the user can solve the simple banking problem using visual programming in the

[14] Note that in Prolog :- denotes reverse implication and ; denotes disjunction.

TABLE 1. Table of bank customers described by different attributes and class 'big spender'.

id	Occupation	Location	Account	Loan	Deposit	Inv. fund	Insur.	Big spender
1	Doctor	Milan	Classic	No	No	TechShare	Family	Yes
2	Doctor	Krakow	Gold	Car	ShortTerm	No	No	Yes
3	Military	Munich	Gold	No	No	No	Regular	Yes
4	Doctor	Catanzaro	Classic	Car	LongTerm	TechShare	Senior	Yes
5	Energy	Poznan	Gold	Apart.	LongTerm	No	No	Yes
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
26	Police	Tarnow	Gold	Apart.	No	No	No	No
27	Nurse	Radom	Classic	No	No	No	Senior	No
28	Education	Catanzaro	Classic	Apart.	No	No	No	No
29	Transport	Warsaw	Gold	Car	ShortTerm	TechShare	Regular	No
30	Police	Cosenza	Classic	Car	No	No	No	No

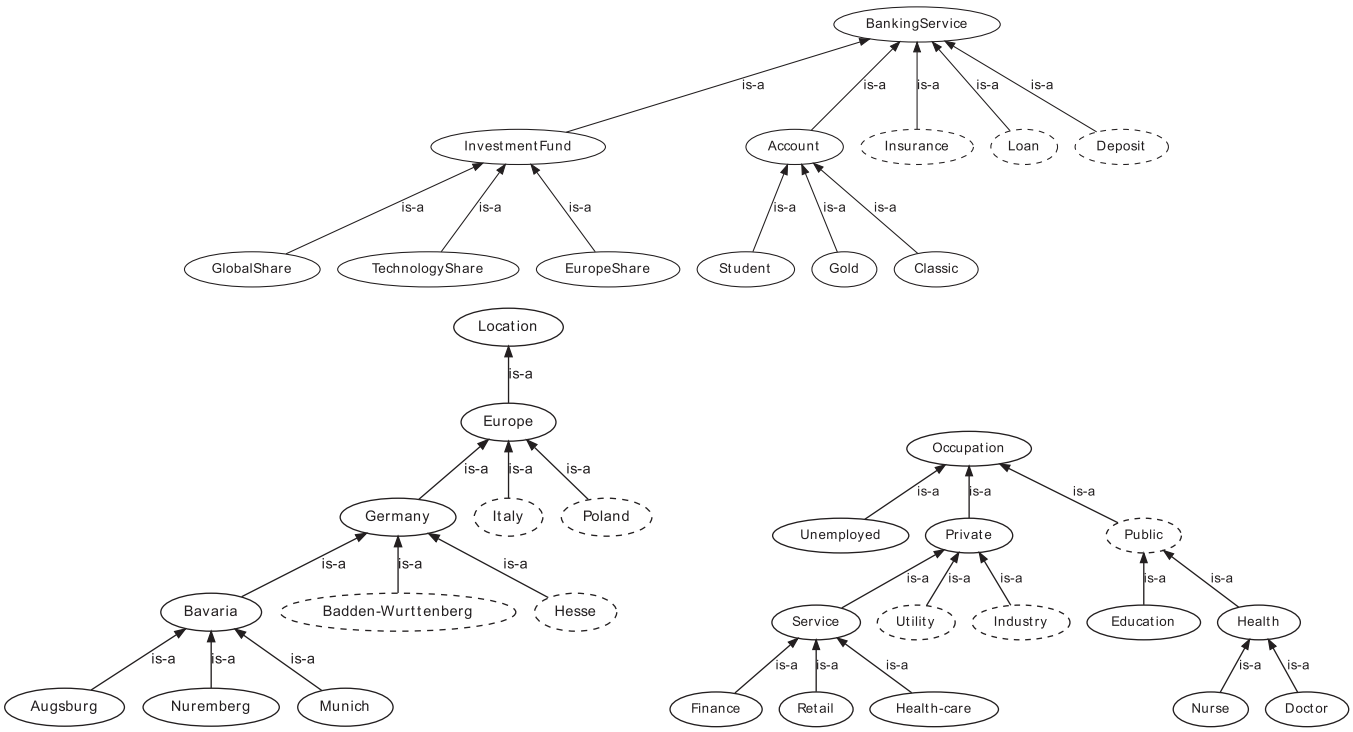


FIGURE 4. The ontologies of banking services, locations and occupations. Concepts with omitted sub-concepts are drawn with a dashed line.

SDM-Toolkit implemented in the service-oriented data mining platform Orange4WS. One of the most important features of Orange, also inherited by Orange4WS (which upgrades Orange to offer the support for SOAP[15] and RESTful[16] web services, which can be used as workflow components), is an easy-to-use interactive workflow construction environment. It enables graphical construction of workflows by allowing workflow elements called *widgets* to be positioned in a desired order, connected with lines representing the flow of data, adjusted by setting their parameters and finally executed. The environment includes a large collection of widgets with various functionalities: data mining and machine learning algorithms, pre-processing and visualization components and others.

The two new semantic subgroup discovery systems presented in this paper have been integrated into Orange4WS forming the SDM-Toolkit which can thus be used to compose workflows for

[15] http://www.w3.org/TR/soap/.

[16] A RESTful web service is a simple web service implemented using HTTP and the principles of REST [26].

```
big_spender(X) ←
   public(X), gold(X).

big_spender(X) ←
   doctor(X), deposit(X).

big_spender(X) ←
   germany(X), service(X), investmentFund(X).
```

FIGURE 5. Three example subgroup descriptions discovered in the banking domain. Each subgroup description represents a set of big spenders.

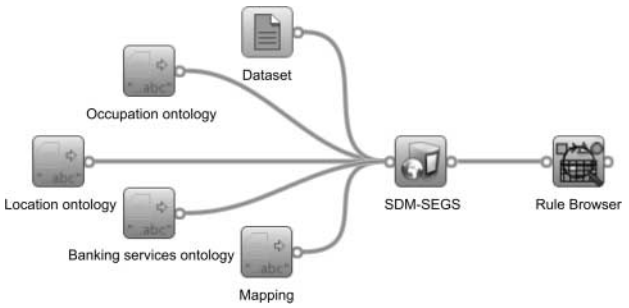
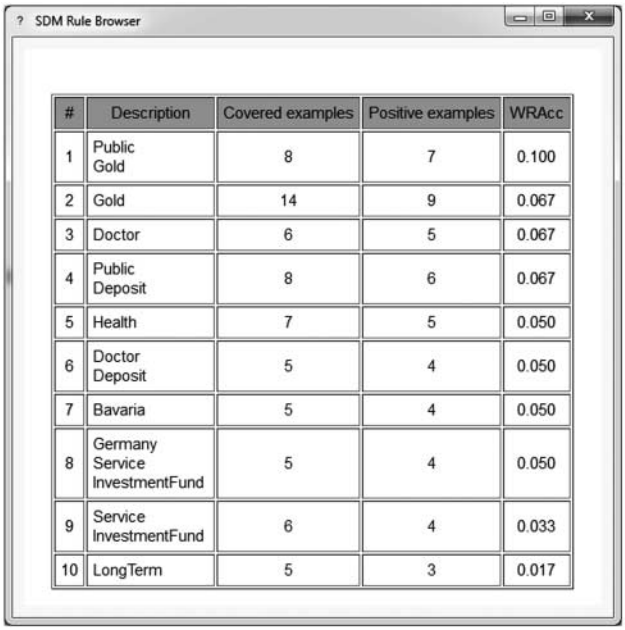


FIGURE 6. A workflow in the SDM-Toolkit for solving the banking problem.

solving various tasks. Figure 6 represents a simple workflow in the SDM-Toolkit. Suppose the user wishes to find some patterns (in the case of SEGS, SDM-SEGS and SDM-Aleph, these are subgroup descriptions) from the given dataset of banking clients and three domain ontologies. First, using the widget denoted as `Dataset`, the user loads the dataset, which can be in various formats (ARFF, CSV, etc.). Next, the user loads the OWL files of the ontologies she wishes to use or simply specifies the URL if the ontology is available on-line. This step can be done using three widgets for reading files into strings. Lastly, using the same type of widget, the user loads the mapping file, which is just a mapping from example identifiers to a list of URIs of ontological concepts. In Fig. 6, the widgets for reading files were renamed appropriately (e.g. `Location Ontology`) for clarity.

The user then connects the output signals of the widgets with the input signals of the widget of the system she wishes to use, in this case we use SDM-SEGS. By double-clicking on the SDM-SEGS widget, the user can fine-tune the desired parameters (e.g. minimum support, maximum rule length, k parameter of WRAcc etc.). The SDM-SEGS output signal can then be connected to the rule browser widget, where the user can scroll through the discovered subgroup descriptions, as shown in Fig. 7.

By swapping the SDM-SEGS widget with SDM-Aleph, the user can solve the task using the SDM-Aleph system instead, whereas SEGS cannot be used for this task because of its domain specificity.



#	Description	Covered examples	Positive examples	WRAcc
1	Public Gold	8	7	0.100
2	Gold	14	9	0.067
3	Doctor	6	5	0.067
4	Public Deposit	8	6	0.067
5	Health	7	5	0.050
6	Doctor Deposit	5	4	0.050
7	Bavaria	5	4	0.050
8	Germany Service InvestmentFund	5	4	0.050
9	Service InvestmentFund	6	4	0.033
10	LongTerm	5	3	0.017

FIGURE 7. Viewing the subgroup descriptions found for the banking problem in the SDM-Toolkit. Each line in a cell in the description column represents one conjunct of the *Conditions* of a given rule.

3.3. Public availability of the SDM-Toolkit

SDM-Toolkit is open-source software licensed under GPL and is publicly available for download at http://kt.ijs.si/software/SDM/. The toolkit contains SDM-SEGS, SDM-Aleph and a widget for browsing rules. SEGS is available for use as a web application at http://kt.ijs.si/software/SEGS/ or together with the SegMine workflow [18], which is available for download at http://segmine.ijs.si. Additionally, a video of constructing an example SDM workflow in Orange4WS (as described in Section 3.2) is available at http://kt.ijs.si/software/SDM/demo.wmv.

4. BIOMEDICAL USE CASES AND EXPERIMENTAL COMPARISON OF SDM ALGORITHMS

In this section, our new systems SDM-SEGS and SDM-Aleph are evaluated and compared with SEGS. Despite the fact that SDM-SEGS and SDM-Aleph are not limited to applications in biology, two such real-life domains are used in our experiments to assess the characteristics of the systems in comparison with the baseline system SEGS whose application is limited to biology (microarray data analysis). This section presents the two domains, the developed reusable workflows implemented in the SDM-Toolkit and a qualitative comparison—supported by experimental results—of SEGS, SDM-SEGS and SDM-Aleph. For the experimental comparison of the systems, we have evaluated the results (rule sets discovered by the three systems) using four main measures for evaluating sets of descriptive rules

proposed by Lavrač *et al.* [24]: the average rule coverage as a measure of generality of the rule set, overall support, average significance of the rule set and average interestingness of the rule set.

4.1. Biomedical use cases

In order to demonstrate the use of the three presented semantic data mining systems for solving real-world problems, we tested the approaches on two publicly available biomedical microarray datasets:

(1) ALL [27] and
(2) hMSC [28]

which we used in our previous research [18]. Both datasets encode gene expression data for two classes. The challenge is to produce descriptions of sets of genes differentially expressed in the given dataset.

The first dataset is a well-known dataset from a clinical trial in ALL, which is a typical dataset for medical research, with several samples available for each class (95 arrays for B-type cells and 33 arrays for T-type cells), where each sample consists of gene expression values for 9001 genes.

The second dataset is known from the analysis of senescence in hMSC. The dataset consists of gene expression profiles from late senescent passages of MSC from three independent donors, together with MSC of early passages. Each sample consists of gene expression values for 20 326 genes.

4.2. Reusable biological workflows in the SDM-Toolkit

Due to the simplicity of the Orange user interface, it is straightforward to devise a workflow for knowledge discovery on the datasets of Section 4.1, and due to the service-oriented functionalities of Orange4WS, the discovery process can be executed in a distributed fashion.

Figure 8 shows an example workflow for solving the described task which performs the pre-processing of raw microarray data, followed by the SDM-SEGS system for discovering the underlying symbolic patterns.

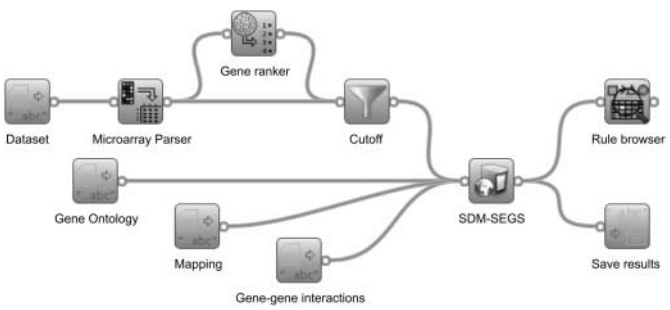


FIGURE 8. A workflow in the SDM-Toolkit for knowledge discovery from microarray data.

```
diff_expressed(X) ←
    'immune system process'(X),
    'plasma membrane'(X),
    interacts(X, Y),
    'T cell receptor signaling pathway'(Y).

diff_expressed(X) ←
    'small molecule metabolic process'(X),
    interacts(X, Y),
    'intracellular membrane-bounded organelle'(Y).

diff_expressed(X) ←
    'anatomical structure morphogenesis'(X),
    'intracellular part'(X),
    'regulation of biological process'(X).
```

FIGURE 9. Selected examples of individual subgroup descriptions discovered by SEGS, SDM-SEGS and SDM-Aleph on the ALL domain, respectively. The predicate names represent ontological concepts and describe a particular set of genes. Each subgroup description represents a set of differentially expressed genes.

The pre-processing steps of knowledge discovery from microarray data, as shown in [18], include *raw data pre-processing* (normalization, missing values removal, merging, etc.), *gene ranking* (e.g. using the ReliefF [29] algorithm) and *filtering out uninteresting genes* (by employing the log FC measure).

These steps are implemented by the following workflow widgets: `Microarray Parser`, `Gene ranker` and `Cutoff`, respectively.

When constructing the workflow, the user can choose any of the systems described in this paper by selecting their corresponding widgets—and in a similar fashion as described in Section 3.2—obtain symbolic descriptions of highly ranked gene sets. Figure 9 presents three example rules discovered by executing the workflow by three systems, SEGS, SDM-SEGS and SDM-Aleph, respectively.

Finally, the user can choose to display the resulting rule set or save the results to an XML file for the possible future re-use.

4.3. Experimental setting

First, we pre-processed the datasets by following the SegMine [18] methodology. Genes were first ranked using the ReliefF [29] algorithm and then filtered using the logarithm of expression fold change (log FC). All genes g with $|\log \mathrm{FC}(g)| < 0.3$ were removed from the set, resulting in 8952 genes in the ALL domain and 11 389 genes in the hMSC domain.

The ranked genes were annotated by GO and KEGG concepts by using the Entrez database to map between gene identifiers and the ontological concepts. Following the approach proposed in [30], the top 300 genes were used as the positive class and from the remaining examples we have randomly selected 300

genes, which were labeled as negative. This selection was done to achieve results comparable between the systems. In practice, one would use full datasets when using SEGS or SDM-SEGS, which have no scalability issues, while according to [30] one should better use a balanced dataset if using ILP methods (like SDM-Aleph) for gene-enrichment analysis. This is in fact due to scalability issues of ILP methods, since in gene-enrichment analysis we have an order of 20 000 ontological concepts. We do not expect such issues if using smaller ontologies.

Both experiments were repeated 20 times where all the three systems were applied on the same two sets (splits) of positive/negative examples. Finally, we have selected the top 20 rules produced by each algorithm, calculated the selected measures and statistically validated the results.

As suggested in [24], we used the following measures:

(1) the *average rule coverage* (COV) measures the average portion of covered examples $n(\text{Cnd}_i)/N$ over a given rule set;
(2) the *overall support* (SUP) is the portion of positive examples covered by the rules, calculated as the true positive rate for the union of subgroups;
(3) the significance measure expresses how much more probable is a given pattern (rule) compared with the expected pattern (default rule), using the likelihood ratio statistic; the *average significance* (SIG) is calculated over a given rule set;
(4) lastly, the *average interestingness* (WRACC) is defined as the average WRAcc of a rule set.

We applied the Friedman test [31] using significance level $\alpha = 0.05$ and the corresponding Nemenyi post-hoc test [32] for each measure separately. This approach is proposed as an alternative to the t-test, which proves to be inappropriate for such a comparison [33].

The Friedman test ranks the algorithms for each split of examples, the best performing algorithm getting the rank of 1, the second best rank 2, etc. In the case of ties, average ranks are assigned. The Friedman test then compares the average ranks of the algorithms. The null-hypothesis states that all the algorithms are equivalent and so their ranks should be equal. If the null-hypothesis is rejected, we can proceed with a post-hoc test, in our case the Nemenyi test. The Nemenyi test is used when we want to compare multiple algorithms to each other. The performance of the algorithms is significantly different if the average ranks differ by at least the *critical distance* (CD).

The visualization of the results, using diagrams is also proposed in [33]. Since the diagrams summarize the results in a compact way, we omit the extensive tables of scores (which were needed for the statistical validation) to avoid clutter and provide tabular results for one quality measure only in Table 2 for illustrative purposes. Table 2 presents a table of achieved scores produced by each algorithm, in this case for the average rule coverage measure.

TABLE 2. Average rule coverage scores for each algorithm for 20 different splits of positive/negative examples.

Split	SEGS	SDM-SEGS	SDM-Aleph
0	0.036	0.097	0.113
1	0.037	0.056	0.104
2	0.036	0.104	0.123
3	0.037	0.106	0.101
4	0.037	0.081	0.105
5	0.041	0.093	0.099
6	0.038	0.095	0.115
7	0.043	0.086	0.114
8	0.036	0.098	0.113
9	0.041	0.061	0.104
10	0.041	0.083	0.123
11	0.037	0.102	0.124
12	0.039	0.084	0.099
13	0.036	0.099	0.106
14	0.038	0.144	0.115
15	0.036	0.111	0.110
16	0.036	0.085	0.104
17	0.037	0.088	0.114
18	0.037	0.087	0.113
19	0.039	0.111	0.109

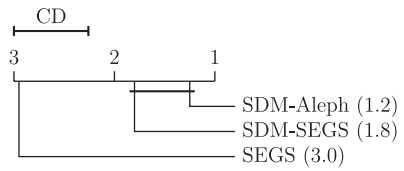


FIGURE 10. Example CD diagram for comparing the algorithms on the hMSC domain for the average support measure, $\alpha = 0.05$.

We produced such tables for each measure, for each of the two domains. These tables were then further analyzed using the Friedman test, which computes the average ranks together with a P-value. If the P-value is lower than our significance level $\alpha = 0.05$, we can reject the null-hypothesis that all the algorithms are equivalent. Then we proceed with the Nemenyi post-hoc test to calculate the CD for the significance level $\alpha = 0.05$, to determine if the difference in the performance between each pair of algorithms is significant. This test can be visualized compactly with a diagram as shown in Fig. 10.

Because we have three algorithms, we first draw the average ranks on the [1, 3] interval. Then we execute the test as follows. If the distance between algorithm A and B is greater than the CD, then we can say that the performance of the better-ranked algorithm is significantly better. Otherwise, if the difference is less than CD, we draw a line between the two algorithms, denoting that we do not have enough evidence to say that one performs significantly better (or worse). Figure 10 is interpreted

TABLE 3. A qualitative comparison of SEGS, SDM-SEGS and SDM-Aleph.

Property	SEGS	SDM-SEGS	SDM-Aleph	Evidence
Domain	Biology	Any	Any	
Ontologies	4	4	Unlimited	
Relations	1	1	Unlimited	
Rule generality (COV)	Low	Medium	High	See Figs 11 and 16
Overall support (SUP)	Low	Medium	High	See Figs 12 and 17
Rule significance (SIG)	High	Medium	Low	See Figs 13 and 18
Cov./prec. trade-off (WRACC)	Low	High	Medium	See Figs 14 and 15

as: SDM-Aleph and SDM-SEGS perform significantly better than SEGS, but there is insufficient evidence to claim that SDM-Aleph performs significantly better than SDM-SEGS.

4.4. Qualitative comparison of SDM-Toolkit subgroup discovery systems

This section provides a qualitative comparison, supported by experimental results, of SEGS, SDM-SEGS and SDM-Aleph, by summarizing the systems' properties and discussing which are the most suitable applications of each system.

Table 3 presents the properties of the presented systems and of the resulting rule sets of each system. The user can be interested in finding rule sets with particular characteristics or has some specific constraints regarding the data to use, depending on the target application of a given system. The user might also wish to use a particular number of ontologies or relations. On the one hand, the user can be interested in more general rules with high support and coverage, as is typical in pattern mining, or on the other hand in specific rules with high significance, as is the case in many biological domains.

With this in mind, we have compared the systems in terms of the following dimensions:

(1) the supported domains;
(2) the number of supported ontologies;
(3) the number of supported relations;
(4) the generality of the resulting rules measured by the average rule coverage (COV);
(5) the overall support of the rule set (SUP);
(6) the average significance of the rule set (SIG) and
(7) the average interestingness of the rule set measured as a trade-off between coverage and precision gain, which is a typical heuristic in subgroup discovery (WRAcc).

The qualitative assessment is supported by the results of experiments in the two biomedical domains.

As mentioned, the SEGS system is domain specific and is limited to four biological ontologies, three sub-parts of the GO and KEGG and supports only one relation between the examples, but provides several biological measures to evaluate

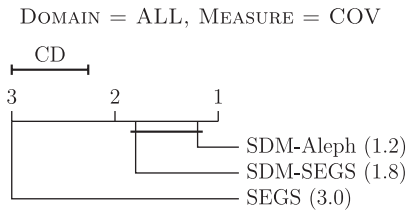


FIGURE 11. CDs between the algorithms on the ALL domain for measure COV, $\alpha = 0.05$.

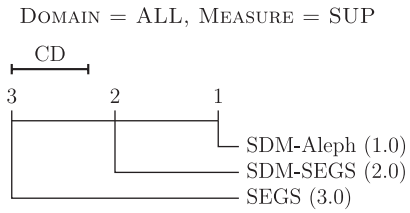


FIGURE 12. CDs between the algorithms on the ALL domain for measure SUP, $\alpha = 0.05$.

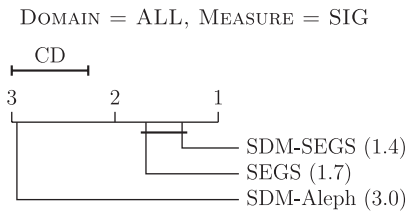


FIGURE 13. CDs between the algorithms on the ALL domain for measure SIG, $\alpha = 0.05$.

the results (mentioned already in Section 2.2). Because of this, the resulting rules tend to be very specific, with high significance, as shown in Figs 11–13.

SDM-SEGS generalizes SEGS so that it is domain independent, enables to import any OWL ontology and uses wWRAcc to select the rules and WRAcc to evaluate the rules, which is a more general purpose evaluation measure. Due to

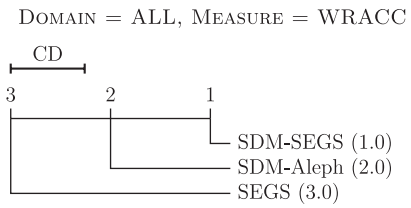


FIGURE 14. CDs between the algorithms on the ALL domain for measure WRACC, $\alpha = 0.05$.

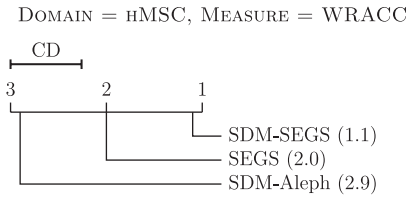


FIGURE 15. CDs between the algorithms on the hMSC domain for measure WRACC, $\alpha = 0.05$.

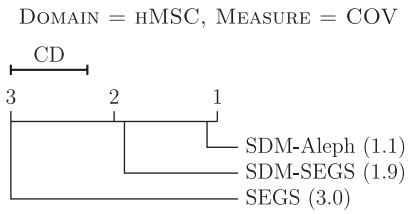


FIGURE 16. CDs between the algorithms on the hMSC domain for measure COV, $\alpha = 0.05$.

this fact, the experimental results show that SDM-SEGS ranks best according to the WRAcc measure (as shown in Figs 14 and 15).

SDM-Aleph has the fewest constraints regarding the input data and also produces the most general rules, with the highest overall support. This is the result of the used rule construction technique, which tends to cover all positive examples.

Figures 11 and 16 show that both SDM-Aleph and SDM-SEGS produce rules with statistically significantly higher coverage, whereas Figs 12 and 17 show that SDM-Aleph and SDM-SEGS cover a significantly higher portion of positive examples than SEGS. Figures 13 and 18 show that the significance of rules of SEGS and SDM-SEGS is on average significantly higher than that of SDM-Aleph. As for the coverage/precision gain trade-off, we can see from Fig. 14, that both SDM-SEGS and SDM-Aleph do significantly better in terms of the WRAcc measure on the ALL domain, whereas on the hMSC domain SDM-SEGS performs significantly better than SEGS. Both SEGS and SDM-SEGS perform significantly better than SDM-Aleph. This indicates that in the case of SDM-Aleph, the WRAcc performance depends on the domain.

In summary, if the user needs a general purpose tool for discovering patterns with high support and coverage, the choice

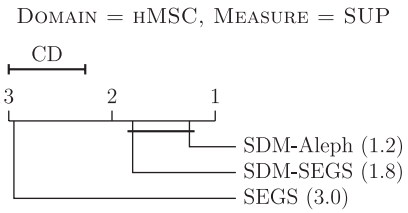


FIGURE 17. CDs between the algorithms on the hMSC domain for measure SUP, $\alpha = 0.05$.

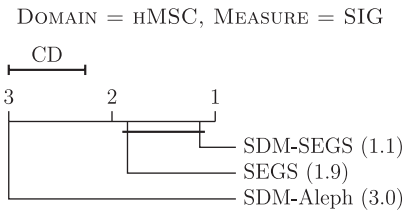


FIGURE 18. CDs between the algorithms on the hMSC domain for measure SIG, $\alpha = 0.05$.

of SDM-Aleph is suggested, otherwise if the user is interested in specific rules, with high significance, she should better choose SDM-SEGS or SEGS in the case of biological domains.

4.5. Runtime comparison of SDM-Toolkit subgroup discovery systems

A few notes on runtime of the three systems are also in place. The runtime was measured on a 64-bit Ubuntu machine with 8 GB of RAM and an Intel i7 processor with 8 cores. On the ALL domain, SDM-Aleph needs on average ∼270 s, whereas SDM-SEGS and SEGS need around 5 and 16 s to complete, respectively. On the hMSC domain, the results are similar, where SDM-Aleph needs around 220 s to complete the execution, while SDM-SEGS and SEGS around 3.5 and 6.5 s, respectively. Figure 19 shows that these differences are all statistically significant.

The time differences are due to the fact that SDM-Aleph's hypothesis language is much more expressive, thus the hypothesis search space grows accordingly, as one can add any number of additional relations and this must be (and is) reflected in Aleph's rule construction algorithm. On the other hand, SDM-SEGS and SEGS exploit the constraints imposed on the hypothesis language (limited number of ontologies and only one relation), resulting in much more time-efficient rule construction.

5. RELATED WORK

This section presents the related work, starting with the work which—like our approach—deals with using taxonomies/ontologies as domain knowledge in learning. As

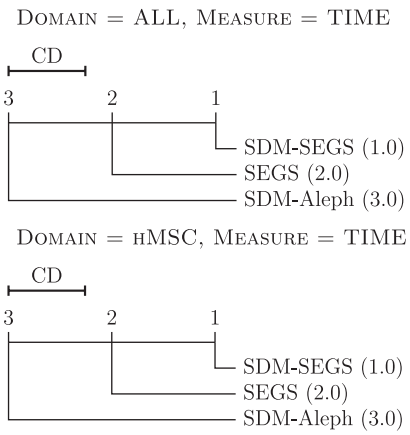


FIGURE 19. CD diagrams for runtime on both domains, $\alpha = 0.05$.

in [34], we divide this work into two main categories. The first category, addressed in Section 5.1, considers taxonomies/ontologies in a standard (relational) learning setting. Together with [16, 34–38], our work fits well into this first category. The second category, outlined in Section 5.2, goes out of the scope of the traditional relational setting, by introducing learning mechanisms into description logics (DLs), hybrid languages integrating Horn logic and DL and learning in a more expressive formalism. This category includes [39–44]. Finally, Section 5.3 covers also some other related work, where other means of using ontologies in the knowledge discovery process are discussed.

5.1. Strongly related work

The most relevant related work is SEGS [16], which has already been thoroughly discussed throughout this paper.

Using taxonomies of predicates to speed up propositionalization, as well as the subsequent step of rule learning using a feature generality taxonomy, is proposed in [34]. The main differences in comparison to our work are that the task they were dealing with is classification and not subgroup discovery and their approach to this was through an intermediate propositionalization step.

In [35], background knowledge is in the standard inheritance network notation and the KBRL[17] algorithm performs a general-to-specific heuristic search for a set of conjunctive rules that satisfy user-defined rule evaluation criteria. Expressiveness of this system is most similar to that of SDM-Aleph and the main difference is in the formalism in which the domain knowledge is encoded. Since there is only a brief description of the algorithm and due to the fact that an implementation is not available, it is difficult to make an experimental comparison.

[17] KBRL is based on the RL learning program of [45].

In [36], the use of taxonomies (the leaves of the taxonomy correspond to attributes of the input data) on paleontological data is studied. The problem was to predict the age of a fossil site on the basis of the taxa that have been found in it; the challenge was to consider taxa at a suitable level of aggregation. Motivated by this application, they studied the problem of selecting an antichain from a taxonomy that improves the prediction accuracy. In contrast to our work, they are interested in classification and do not consider additional relations between the examples.

In [37], an engineering ontology of computer-aided design (CAD) elements and structures is used as background knowledge to extract frequent product design patterns in CAD repositories and discovering predictive rules from CAD data.

Using a data mining ontology for meta-learning has been proposed in [38]. In meta-learning, the task is to use data mining techniques to improve base-level learning. The data mining ontology is used to (1) incorporate specialized knowledge of algorithms, data and workflows and to (2) structure the search space when searching for frequent patterns.

5.2. Weakly related work

The most commonly used DL format for semantic web is OWL-DL. OWL-DL allows to define properties of relations which link entities defined in an ontology as transitive, symmetric, functional and to assign cardinality to relations. Properties of relations form an important part of the domain knowledge model, therefore modifications of existing relational algorithms or even new algorithms are required in order to effectively exploit this knowledge.

Kietz [39] was one of the first to make a step in this direction by extending the standard learning bias used in ILP with DL (CARIN-$\mathcal{ALN}$).

More recently, Lehmann and Haase [40] defined a refinement operator in the $\mathcal{EL}$ DL; opposed to our work they consider only the construction of consistent and complete hypotheses using an ideal refinement operator. Furthermore, in contrast with their work, this paper discusses mostly subgroup discovery. In addition, the hypothesis language in their approach are expressions in $\mathcal{EL}$, while we use Horn clauses as the hypothesis language.

In [41], they introduce an algorithm named Fr-ONT for frequent concept mining expressed in $\mathcal{EL}^{++}$ DL. In contrast to our work, the task they are solving is frequent concept mining and the hypothesis language they are using is $\mathcal{EL}^{++}$ DL.

Combining web mining and the semantic web was proposed in [42]. The initial work in that direction includes [43, 44], where the authors propose an approach to mining the semantic web by using a hybrid language $\mathcal{AL}$-log, which allows a unified treatment of structural and relational features of data by combining $\mathcal{ALC}$ and DATALOG. In their proposal, this framework was developed for mining multi-level association rules and not subgroup discovery.

5.3. Other work

In [46], ontology-enhanced association mining is discussed and four stages of the (4ft-Miner-based) KDD process are identified that are likely to benefit from ontology application: data understanding, task design, result interpretation and result dissemination over the semantic web.

The work of Brisson and Collard [47] first focuses on pre-processing steps of business and data understanding in order to build an ontology-driven information system, and then the knowledge base is used for the post-processing step of model interpretation. In [20], Liu proposes a learning-based semantic search algorithm to suggest appropriate semantic web terms and ontologies for the given data.

An ontology-driven approach to knowledge discovery in biomedicine is described in [48], where efforts to bridge knowledge discovery in biomedicine and ontology learning for successful data mining in large databases are presented.

6. CONCLUSIONS

This paper addresses semantic data mining, a new data mining paradigm in which ontologies are exploited in the process of data mining and knowledge discovery.

We present the SDM-Toolkit that enables the user to exploit ontologies in the process of data mining and knowledge discovery. Our toolkit is implemented in the service-oriented data mining platform OrangeWS and is made publicly available for download.

The set of tools presented in this paper includes three semantic subgroup discovery systems: SEGS, a successful domain-specific system for analyzing microarray data and two new general-purpose systems SDM-SEGS and SDM-Aleph. We demonstrate how to use our tools on a simple example and on two advanced real-world biomedical case studies. We provide a qualitative comparison of the developed systems, based on their extensive experimental evaluation, while a thorough biological interpretation of the resulting rules is beyond the scope of this paper.

In this work, we have exploited only a limited amount of power offered by RDF/OWL technologies. In further work, we plan to investigate how to further exploit these technologies for data mining. One can imagine having additional information about the characteristics of the data attributes themselves, for instance, information about the uncertainty of an attribute, how does a certain attribute relate to some other attribute or how to use an attribute (e.g. for automatically using temporal or spatial information).

In further work, we plan to develop a fast system for mining an arbitrary number of relations and ontologies, which will exploit as much as possible the vast range of functionalities offered by the OWL family of languages. In addition, our plan is to investigate the possibility of applying the presented methods to mining-linked open data or, if the existing algorithms prove not to be sufficiently effective in this challenging new setting, to propose new semantic data mining algorithms.

An important part of our further work will also be adding additional algorithms into SDM-Toolkit for solving other data mining tasks (e.g. decision tree learning using ontological background knowledge), as well as presenting a general mechanism for transforming a data mining algorithm into a semantic data mining algorithm.

ACKNOWLEDGEMENTS

We wish to thank Vitor Santos Costa for his idea of using tabling in SDM-Aleph to improve its performance, as well as to Petra Kralj Novak, Igor Trajkovski, Larisa Soldatova and Vid Podpečan for fruitful discussions and collaboration in the development of the SEGS algorithm and the Orange4WS data mining environment.

FUNDING

This work was supported by the Slovenian Ministry of Higher Education, Science and Technology [grant number P-103] and the EU FP7 project e-LICO.

REFERENCES

[1] Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996) From data mining to knowledge discovery in databases. *AI Mag.*, **17**, 37–54.

[2] Witten, I.H., Frank, E. and Hall, M.A. (2011) *Data Mining: Practical Machine Learning Tools and Techniques* (3rd edn). Morgan Kaufmann Publishers, San Francisco, CA, USA.

[3] Berthold, M.R., Cebron, N., Dill, F., Gabriel, T.R., Kötter, T., Meinl, T., Ohl, P., Sieb, C., Thiel, K. and Wiswedel, B. (2007) KNIME: The Konstanz Information Miner. *Proc. Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*, Freiburg, Germany, March 7–9, pp. 319–326. Springer, Berlin/Heidelberg, Germany.

[4] Demšar, J., Zupan, B., Leban, G. and Curk, T. (2004) Orange: From Experimental Machine Learning to Interactive Data Mining. *Proc. 8th European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD'04)*, Pisa, Italy, September 20–24, pp. 537–539. Springer, Berlin/Heidelberg, Germany.

[5] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M. and Euler, T. (2006) YALE: Rapid Prototyping for Complex Data Mining Tasks. *Proc. 12th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'06)*, Philadelphia, PA, USA, August 20–23, pp. 935–940. ACM Press, NY, USA.

[6] Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)* (1st edn). Morgan Kaufmann Publishers, San Francisco, CA, USA.

[7] Lloyd, S. (1982) Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, **28**, 129–137.

[8] Agrawal, R. and Srikant, R. (1994) Fast Algorithms for Mining Association Rules in Large Databases. *Proc. 20th Int. Conf. Very Large Data Bases (VLDB'94)*, Santiago de Chile, Chile, September 12–15, pp. 487–499. Morgan Kaufmann Publishers, San Francisco, CA, USA.

[9] Klösgen, W. (1996) *Advances in Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence, Menlo Park, CA, USA.

[10] Wrobel, S. (1997) An Algorithm for Multi-relational Discovery of Subgroups. *Proc. 1st European Conf. Principles of Data Mining and Knowledge Discovery (PKDD'97)*, Trondheim, Norway, June 24–27, pp. 78–87. Springer, Berlin, Germany.

[11] Muggleton, S. (ed.) (1992) *Inductive Logic Programming*. Academic Press, London.

[12] De Raedt, L. (2008) *Logical and Relational Learning*. Springer, Berlin/Heidelberg, Germany.

[13] Džeroski, S. and Lavrač, N. (eds) (2001) *Relational Data Mining*. Springer, Berlin.

[14] Železný, F. and Lavrač, N. (2006) Propositionalization-based relational subgroup discovery with RSD. *Mach. Learn.*, **62**, 33–63.

[15] Podpečan, V., Zemenova, M. and Lavrač, N. (2011) Orange4WS environment for service-oriented data mining. *Comput. J.*, Online access. Advanced Access Published 7 August 2011: 10.1093/comjnl/bxr077.

[16] Trajkovski, I., Lavrač, N. and Tolar, J. (2008) SEGS: search for enriched gene sets in microarray data. *J. Biomed. Inf.*, **41**, 588–601.

[17] Lavrač, N., Vavpetič, A., Soldatova, L., Trajkovski, I. and Kralj Novak, P. (2011) Using Ontologies in Semantic Data Mining with SEGS and g-SEGS. *Proc. Int. Conf. Discovery Science (DS'11)*, Espoo, Finland, October 5–7, pp. 165–178. Springer, Berlin/Heidelberg, Germany.

[18] Podpečan, V. *et al*. (2011) SegMine workflows for semantic microarray data analysis in Orange4WS. *BMC Bioinf.*, **12**, 416.

[19] Lavrač, N., Novak, P., Mozetič, I., Podpečan, V., Motaln, H., Petek, M. and Gruden, K. (2009) Semantic Subgroup Discovery: Using Ontologies in Microarray Data Analysis. *Proc. Annual Int. Conf. IEEE, Engineering in Medicine and Biology Society (EMBC'09)*, Minneapolis, USA, September 2–6, pp. 5613–5616. Institute of Electrical and Electronics Engineers, New York, USA.

[20] Liu, H. (2010) Towards Semantic Data Mining. *Doctoral Consortium of the 9th Int. Semantic Web Conf. (ISWC'10)*, Shanghai, China, November 7–11. http://data.semanticweb.org/conference/iswc/2010/paper/448.

[21] Subramanian, A. *et al*. (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl Acad. Sci. USA*, **102**, 15545–15550.

[22] Kim, S.Y. and Volsky, D. (2005) PAGE: parametric analysis of gene set enrichment. *BMC Bioinf.*, **6**, 144–155.

[23] Khatri, P. and Drăghici, S. (2005) Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, **21**, 3587–3595.

[24] Lavrač, N., Kavšek, B., Flach, P. and Todorovski, L. (2004) Subgroup discovery with CN2-SD. *J. Mach. Learn. Res.*, **5**, 153–188.

[25] Muggleton, S. (1995) Inverse entailment and progol. *New Gener. Comput.*, Special issue on Inductive Logic Programming, **13**, 245–286.

[26] Fielding, R. (2000) Architectural styles and the design of network-based software architectures. PhD Thesis.

[27] Chiaretti, S., Li, X., Gentleman, R., Vitale, A., Vignetti, M., Mandelli, F., Ritz, J. and Foa, R. (2004) Gene expression profile of adult t-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. *Blood*, **103**, 2771–2778.

[28] Wagner, W. *et al*. (2008) Replicative senescence of mesenchymal stem cells: a continuous and organized process. *PLoS ONE*, **3**, e2213.

[29] Robnik-Šikonja, M. and Kononenko, I. (2003) Theoretical and empirical analysis of ReliefF and RReliefF. *Mach. Learn.*, **53**, 23–69.

[30] Trajkovski, I., Železný, F., Lavrač, N. and Tolar, J. (2008) Learning relational descriptions of differentially expressed gene groups. *IEEE Trans. Syst. Man Cybern. C*, **38**, 16–25.

[31] Friedman, M. (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.*, **32**, 675–701.

[32] Nemenyi, P.B. (1963) Distribution-free multiple comparisons. PhD Thesis.

[33] Demšar, J. (2006) Statistical comparison of classifiers over multiple data sets. *J. Mach. Learn. Res.*, **7**, 1–30.

[34] Žáková, M. and Železný, F. (2007) Exploiting Term, Predicate, and Feature Taxonomies in Propositionalization and Propositional Rule Learning. *Proc. 18th European Conf. Machine Learning and the 11th European Conf. Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD'07)*, Warsaw, Poland, September 17–21, pp. 798–805. Springer, Berlin/Heidelberg, Germany.

[35] Aronis, J., Provost, F. and Buchanan, B. (1996) Exploiting Background Knowledge in Automated Discovery. *Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining (KDD'96)*, Portland, OR, USA, August 2–4, pp. 355–358. AAAI Press, Menlo Park, CA, USA.

[36] Garriga, G., Ukkonen, A. and Mannila, H. (2008) Feature Selection in Taxonomies with Applications to Paleontology. *Proc. 11th Int. Conf. Discovery Science (DS'08)*, Budapest, Hungary, October 13–16, pp. 112–123. Springer, Berlin/Heidelberg, Germany.

[37] Žáková, M., Železný, F., García-Sedano, J.A., Tissot, C.M., Lavrač, N., Kremen, P. and Molina, J. (2006) Relational Data Mining Applied to Virtual Engineering of Product Designs. *Proc. 16th Int. Conf. Inductive Logic Programming (ILP'06)*, Santiago de Compostela, Spain, August 24–27, pp. 439–453. Springer, Berlin/Heidelberg, Germany.

[38] Hilario, M., Nguyen, P., Do, H., Woznica, A. and Kalousis, A. (2011) *Meta-Learning in Computational Intelligence*. Springer, Berlin/Heidelberg, Germany.

[39] Kietz, J.U. (2002) Learnability of Description Logic Programs. In Matwin, S. and Sammut, C. (eds), *Proc. 12th Int. Conf. Inductive Logic Programming (ILP'02)*, Sidney, Australia, pp. 117–132. Springer, Heidelberg, Germany.

[40] Lehmann, J. and Haase, C. (2009) Ideal Downward Refinement in the EL Description Logic. *Proc. 19th Int. Conf. Inductive Logic*

Programming (ILP'09), Leuven, Belgium, July 2–4, pp. 73–87. Springer, Berlin/Heidelberg, Germany.

[41] Lawrynowicz, A. and Potoniec, J. (2011) Fr-ONT: An Algorithm for Frequent Concept Mining with Formal Ontologies. *Foundations of Intelligent Systems—19th Int. Symp. (ISMIS'11)*, Warsaw, Poland, June 28–30, pp. 428–437. Springer, Berlin/Heidelberg, Germany.

[42] Berendt, B., Hotho, A. and Stumme, G. (2002) Towards Semantic Web Mining. *Proc. Int. Semantic Web Conf. (ISWC'02)*, Sardinia, Italy, June 9–12, pp. 264–278. Springer, Berlin/Heidelberg, Germany.

[43] Lisi, F.A. and Malerba, D. (2004) Inducing multi-level association rules from multiple relations. *Mach. Learn.*, **55**, 175–210.

[44] Lisi, F. and Esposito, F. (2005) Mining the Semantic Web: A Logic-Based Methodology. *Foundations of Intelligent Systems*, pp. 437–440. Springer, Berlin/Heidelberg, Germany.

[45] Clearwater, S. and Provost, F. (1990) Rl4: A Tool for Knowledge-Based Induction. *Proc. 2nd Int. IEEE Conf. Tools for Artificial Intelligence (ICTAI'90)*, Herndon, VA, USA, November 6–9, pp. 24–30. IEEE Computer Society Press, Washington, USA.

[46] Svátek, V., Rauch, J. and Ralbovský, M. (2005) Ontology-Enhanced Association Mining. *Proc. Semantics, Web and Mining, Joint Int. Workshops (EWMF'05 and KDO'05)*, Porto, Portugal, October 3, pp. 163–179. Springer, Berlin/Heidelberg, Germany.

[47] Brisson, L. and Collard, M. (2008) How to Semantically Enhance a Data Mining Process? *Proc. 10th Int. Conf. Enterprise Information Systems (ICEIS'08)*, Barcelona, Spain, June 12–16, pp. 103–116. Springer, Berlin/Heidelberg, Germany.

[48] Gottgtroy, P., Kasabov, N. and MacDonell, S. (2004) An Ontology Driven Approach for Knowledge Discovery in Biomedicine. *Proc. 8th Pacific Rim Int. Conf. Artificial Intelligence (PRICAI'04)*, Auckland, New Zealand, August 9–13. Springer, Berlin/Heidelberg, Germany.