SEMANTIC SUBGROUP DISCOVERY

Anže Vavpetič

Doctoral Dissertation
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Supervisor: Prof. Dr. Nada Lavrač, Jožef Stefan Institute, Ljubljana, Slovenia, and Jožef Stefan International Postgraduate School, Ljubljana, Slovenia

Evaluation Board:
Prof. Dr. Sašo Džeroski, Chair, Jožef Stefan Institute, Ljubljana, Slovenia, and Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
Prof. Dr. Ljupčo Todorovski, Member, Faculty of Administration, University of Ljubljana, Slovenia
Prof. Dr. Filip Železný, Member, Czech Technical University, Prague, Czech Republic

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL

Anže Vavpetič

SEMANTIC SUBGROUP DISCOVERY

Doctoral Dissertation

SEMANTIČNO ODKRIVANJE PODSKUPIN

Doktorska disertacija

Supervisor: Prof. Dr. Nada Lavrač

Ljubljana, Slovenia, March 2016

To Kaja and Črt

Acknowledgments

This thesis would never be finished without the help of a number of people. Their support and help throughout my studies is greatly appreciated.

First of all I would like to thank my supervisor Nada Lavrač for her infinite amount of patience, enthusiasm, support and her always helpful and insightful research ideas and comments. A big thank you also to the members of my evaluation board Sašo Džeroski, Ljupčo Todorovski and Filip Železný for reading my thesis and providing comments for improvement.

I also wish to thank the funding bodies that financially supported my research: the Slovenian Research Agency, the Department of Knowledge Technologies at the Jožef Stefan Institute, the Jožef Stefan International Postgraduate School, and the European Commission for funding the research projects e-LICO, ConCreTe and HBP in which I was involved.

I am also very grateful to Petra Kralj Novak and Igor Trajkovski for sparking the idea that eventually became this dissertation. Big thanks also to everyone I had the pleasure co-authoring papers with, especially: Prem Raj Adhikari, Jakko Hollmén, Vid Podpečan and Senja Pollak. Thanks to Nicolas Lachiche and his team for contributing to my relational data mining package.

I am thankful to all the work colleagues at the Department of Knowledge Technologies for a really friendly and welcoming environment. Special thanks goes to Jan Kralj, Janez Kranjc and Matic Perovšek for the endless discussions about nothing and everything. I am endlessly grateful to Mili Bauer and Tina Anžič for always being prepared to help with the intricacies of working at the Jožef Stefan Institute, as well as to Vid Podpečan, Borut Sluban, Matjaž Juršič and Miha Grčar for always having time for a friendly talk.

Last but not least, I must thank my family and friends for their unending encouragement and support. Finally, I am most grateful to my wife Kaja for her love, understanding, support and for believing in me even at the most difficult times. Thanks also to Črt for letting his dad sleep when he needed it most.

Abstract

The thesis addresses semantic subgroup discovery (SSD), a task at the intersection of relational data mining and semantic web technologies. While subgroup discovery involves finding statistically most interesting population subgroups (e.g., subgroups that are large and have unusual statistical characteristics with respect to some property of interest), in SSD we exploit ontological and other structural background knowledge to improve the subgroup discovery process.

We developed a formal framework for semantic subgroup discovery and illustrated the approach on a motivating example. We designed and implemented new algorithms, which we applied and evaluated in several settings. The new approaches are implemented in SSD algorithms SDM-SEGS, SDM-Aleph, and Hedwig, which were systematically evaluated using a number of evaluation measures on two microarray datasets. In statistical validation, Hedwig proved to be the most successful, followed closely by the others, as no approach dominated in terms of all the evaluation measures. The developed software is open-source and available as Python packages, as well as widgets in the ClowdFlows data mining platform.

Our approaches were applied to three real-life applications: explaining subgroups of breast cancer patients, multi-resolution 0–1 analysis of DNA aberration data, and subgroup discovery on financial news articles. In the first application we used the Gene Ontology as background knowledge used in SSD to generate explanations of patient subgroups; the approach was compared to the supporting factors methodology. Our results agreed with a previous study of breast cancer grades. Furthermore, the methodology was made available as a ClowdFlows workflow, making the experiment repeatable and easily adaptable to similar problems. In the DNA aberration study, SSD was part of a three-part methodology, together with clustering through mixture modeling and banded matrices used for innovative cluster and rule visualization. The methodology provided novel insights on DNA aberration data. Furthermore, the approach was applied on four publicly available datasets, together with background knowledge collected from DBpedia. The methodology proved useful on all but one dataset, where the data did not demonstrate any banded structure at all.

In the financial news application, we wanted to gain insight into a vast collection of news articles; more specifically, we investigated the relationship between the financial market perception of a financial entity and the articles mentioning it. We chose Portugal as the target financial entity, together with a semi-automatically constructed ontology. To model the market perception we used the credit default swap (CDS) price, reflecting the probability that a country will be unable to repay its debt. Using Hedwig we found two interesting rules describing the local extremes in the CDS price.

The thesis also contributes significantly to the field of relational data mining (RDM); we developed a Python library and widgets for the data mining platform ClowdFlows. The package includes a number of RDM approaches, as well as support for MySQL and PostgreSQL, all through a simple API. The package aims to alleviate many of the issues a researcher new to RDM might encounter.

Povzetek

Disertacija obravnava semantično odkrivanje podskupin (SOP), področje na presečišču relacijskega podatkovnega rudarjenja in semantičnega spleta. Naloga odkrivanja podskupin obsega iskanje statistično najbolj zanimivih podskupin v populaciji (npr. podskupine, ki so čim večje in imajo najbolj nenavadne statistične značilnosti v primerjavi z neko izbrano lastnostjo). V okviru SOP izkoriščamo ontološke in druge strukturirane oblike predznanja za izboljšanje procesa odkrivanja podskupin.

Razvili smo formalni okvir za semantično odkrivanje podskupin in pristop ilustrirali na motivacijskem primeru. Zasnovali smo nove algoritme, ki smo jih uporabili in ocenili na več domenah. Nove implementacije SOP pristopov SDM-SEGS, SDM-Aleph in Hedwig smo sistematično primerjali na dveh mikromrežah s pomočjo različnih mer za ocenjevanje podskupin. Statistična validacija je pokazala, da se najbolje odreže pristop Hedwig, ki mu tesno sledijo ostali, saj noben pristop ni prevladoval v vseh merilih ocenjevanja. Razviti programi so odprtokodni in na voljo v obliki paketov za programski jezik Python, ter v obliki gradnikov za platformo ClowdFlows za podatkovno rudarjenje.

Pristope smo uporabili na treh praktičnih primerih: razlaganje podskupin obolelih za rakom na prsih, analiza podatkov o DNK aberacijah in odkrivanje podskupin iz finančnih novic. Na prvem praktičnem primeru smo s pomočjo SOP uporabili ontologijo genov (angl. Gene Ontology) za generiranje razlag podskupin obolelih. Pristop smo primerjali z metodologijo podpornih faktorjev. Naši rezultati so se ujemali s predhodno raziskavo. Metodologijo smo pripravili tudi v obliki delotoka v platformi ClowdFlows, kar omogoča ponovljivost eksperimenta, prav tako pa je mogoče delotok enostavno prilagoditi reševanju podobnih problemov. V okviru raziskave analize podatkov o DNK aberacijah je bilo SOP eno od treh delov metodologije, ki je vključevala še mešane modele (angl. mixture models) ter pasovne matrike. Metodologija vse tri pristope združuje v inovativno vizualizacijo za gruče in pravila. Pristop je ponudil nov vpogled v podatke o DNK aberacijah. Metodologijo smo uporabili tudi na štirih javno dostopnih zbirkah podatkov, ki vključujejo tudi predznanje, zbrano z DBpedije. Z izjemo ene zbirke podatkov, ki ni izkazala lastnosti pasovnih matrik, se je pristop izkazal kot primeren.

Pri finančnem praktičnem primeru smo hoteli pridobiti vpogled v širok nabor novic. Bolj natančno smo raziskovali razmerje med percepcijo finančnega trga neke entitete in članke, ki jo omenjajo. Za ciljno finančno entiteto smo izbrali Portugalsko. Prav tako smo uporabili pol-avtomatsko izdelano ontologijo. Za model percepcije trga smo uporabili ceno zamenjave kreditnega tveganja (angl. credit default swap), ki odraža verjetnost, da država ne bo mogla odplačati dolga. Z uporabo Hedwig smo našli dve zanimivi pravili, ki opisujeta lokalne ekstreme v ceni zamenjave kreditnega tveganja.

Disertacija pomembno prispeva tudi k področju relacijskega podatkovnega rudarjenja (RPP). Razvili smo knjižnico za programski jezik Python in nabor gradnikov za platformo ClowdFlows. Paket vključuje številne pristope za RPP in podporo za MySQL in PostgreSQL prek preprostega programskega vmesnika. Paket poskuša olajšati čim več težav, ki bi jih raziskovalec, ki se je šele začel ukvarjati z RPP, utegnil srečati pri delu.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

ALL ... Acute Lymphoblastic Leukemia
API ... Application Programming Interface
DAG ... Directed Acyclic Graph
DL ... Description Logics
DM ... Data Mining
FDR ... False Discovery Rate
FOL ... First-order Logic
FWER .. Familywise Error Rate
GO ... Gene Ontology
ILP ... Inductive Logic Programming
KDD ... Knowledge Discovery in Databases
KEGG .. Kyoto Encyclopedia of Genes and Genomes orthology
ML ... Machine Learning
OWL ... Web Ontology Language
RDF ... Resource Description Format
RDBS ... Relational Database Management System
RDM ... Relational Data Mining
RSD ... Relational Subgroup Discovery
SDM ... Semantic Data Mining
SEGS ... Search for Enriched Gene Sets
SSD ... Semantic Subgroup Discovery

Chapter 1

Introduction

This thesis presents a formal framework for semantic subgroup discovery (SSD) as well as new semantic subgroup discovery algorithms, applied and evaluated in several scenarios. The developed algorithms and methodology are open source and have corresponding components in the ClowdFlows data mining platform. This enables reuse, sharing and extendibility of the developed approaches. Along with SSD algorithms developed in this thesis, the implementation also includes several popular inductive logic programming (ILP), relational data mining (RDM), and propositionalization algorithms.

1.1 Problem Description

Knowledge discovery in databases (KDD) refers to the interactive and iterative process of finding interesting patterns and models in data [2]. The most common setting in knowledge discovery is rather simple: given is the empirical data and a data mining task to be solved. First, the data is preprocessed, 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 and interpreted.

In this thesis we are interested in symbolic data analysis techniques, which aim at finding comprehensible patterns/models in the data. Decision tree learning [3], [4] and classification rule learning [5], [6] are popular examples of methods, which aim at building models from class labeled data, enabling classification of yet unseen examples. In contrast to classification rule learning, descriptive rule learning, which focuses on learning from unlabeled data, aims to find descriptive sets of patterns describing the data [7]. In the thesis we focus mainly on supervised descriptive rule learning [8], a task at the intersection of classification rule learning and descriptive rule learning, where the goal is to induce descriptive patterns from class labeled data.

Early rule learning algorithms [9] have focused on learning classification rules from tabular data. Best known examples of these algorithms are AQ [10], CN2 [5], and Ripper [6]. Relational rule learning [11], on the other hand, takes as input a set of tables or a multi-relational database, and results in a set of relational rules, expressed as a logic program [12] or some other relational formalism. It is well known from the literature on RDM [11] and ILP [13], [14] 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, are ontologies [15]. An ontology defines a set of representational primitives to model a domain of knowledge and can act as a mean of providing additional information to machine learning (data mining) algorithms by attaching semantic descrip-

tors 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 representation which encourages knowledge reuse. Two popular formats are the Web Ontology Language (OWL) for ontologies, which is built on top of the Resource Description Framework (RDF). This domain knowledge is usually built collaboratively by domain experts.

In data mining experiments there is typically 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 work uses the term semantic data mining [16], [17] 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 defined by domain experts and can make symbolic patterns even more comprehensible. Moreover, in rule learning, using higher-level ontological concepts provides the means for more effective generalizations which would not have been possible by using only the terms used in instance descriptions.

In this work we focus on the problem of semantic subgroup discovery, which has not been addressed in the related work so far. One step in this direction was made in [18] with the system SEGS. SEGS addresses the task of searching for enriched gene sets and cannot be directly used for solving general subgroup discovery tasks. Furthermore, we wish to exploit ontological background knowledge, since this provides us with several advantages: automatically inducing generalizations that standard algorithms could not make; search space pruning based on the ontological hierarchy of concepts; using semantic subgroup discovery to produce explanations of subgroups via vocabulary from the domain ontology.

1.2 Hypotheses

Hypothesis 1 The drawback of current data mining tools, which can be used for subgroup discovery and are able to include background knowledge in the induction process (e.g., Aleph) is that the background knowledge is typically not represented in a uniform and standardized format. Our hypothesis is that domain knowledge in a standard language such as RDF or OWL is suitable to be used as background knowledge in data mining, since the languages promote re-usability and tools for cooperative and formal development of domain knowledge.

Hypothesis 2 Our main hypothesis is that the effectiveness of data mining algorithms can be improved (i.e. achieving better generalizations) by considering the relations between attribute-values encoded in the domain knowledge structure (e.g., an ontology that provides additional knowledge about attribute-values of the input examples).

1.3 Objectives and Contributions

There are several objectives of the dissertation, which led to particular scientific advances listed below.

Objective 1 Improve the accessibility and compatibility of existing ILP and RDM approaches.

Contribution 1 We developed an open-source `python-rdm` library and a new ClowdFlows package including most of contemporary ILP and RDM algorithms. Algorithm availability alleviates much of the issues a researcher new to these fields might encounter, given that we provide a common interface to several algorithms, including the popular ILP system Aleph together with its feature construction component, as well as RSD, RelF and Wordification propositionalization engines. The package has also external contributions in the form of several RDM systems: Tertius, 1BC, 1BC2, Cardinalization, Quantiles and Relaggs. The package has support for MySQL and PostgreSQL databases. This is not a core scientific contribution of the thesis, but rather a technological advancement, contributing to open science through easier access to algorithms in the area. The library and the ClowdFlows package source code are freely available at `https://github.com/xflows/rdm`.

The results addressing this objective, described in Chapter 3, were published in a journal paper [19] and a conference paper [20].

Objective 2 Devise a theoretical framework for semantic subgroup discovery and investigate the applicability of existing ILP and RDM approaches for semantic subgroup discovery. Based on the comparison of existing algorithms, propose a general-purpose method that takes the best of those two worlds.

Contribution 2.1 We have established a unifying semantic subgroup discovery framework, including a task definition grounded on subgroup discovery and relational data mining and detailed elaboration on how a domain ontology fits into this picture.

Contribution 2.2 We have adapted two existing approaches SEGS and Aleph, resulting in SDM-SEGS and SDM-Aleph algorithms, respectively. The approaches, which were shown to be generally applicable on a motivational banking domain, were compared to the state of the art SEGS algorithm on two biomedical microarray datasets (acute lymphoblastic leukemia and human mesenchymal stem cells) and evaluated using several subgroup discovery measures. While the results show that there is no absolute winner (i.e. dominating in all metrics), SDM-SEGS proves to be orders of magnitude faster, while still achieving good results. SDM-Aleph produces rules with the best coverage and support, but is also the slowest.

Contribution 2.3 We developed a new general-purpose semantic subgroup discovery algorithm which can take general ontologies as input to semantic rule construction.

We consider this development the *main scientific contribution* of the thesis.

The results addressing this objective described in Chapter 4, were published in two journal articles [21], [24], and two conference papers [22], [23].

Objective 3 Provide reference implementations of developed software available as an open source toolbox for semantic subgroup discovery.

Contribution 3 Our software contributions are all open-source and available to be used in two ways: as libraries or web services to be used programatically, or as components in the ClowdFlows data mining platform.

Similar to Contribution 1, this is not a core scientific contribution of the thesis, but rather a technological advancement, contributing to open science through easier access to algorithms in the area.

The details of the available software are described in Chapter 4.

Objective 4 Investigate the utility of ontological domain knowledge in improving the process of data mining, and discover the potential benefits and drawbacks of the developed approaches on real-life problem domains.

Contribution 4.1 We developed a general-purpose methodology called Explain-SD, where semantic subgroup discovery is employed to use ontological background knowledge as a vocabulary for describing patterns resulting from standard data mining techniques (e.g., clusters, subgroups, subgraphs). We showed that such higher-level descriptions have the potential to provide new insights into the domain of investigation, and that this can be ensured by using semantic subgroup discovery methods. We demonstrated this on a motivating use case and on a gene expression profiling use case where groups of breast cancer patients, identified through subgroup discovery in terms of gene expression, are further explained through concepts from the Gene Ontology and KEGG orthology. The methodology was compared to the supporting factors technique for characterizing subgroups.

We consider this result, published in a conference paper [25] and a journal article [26], to be one of the *main scientific contributions* of the thesis.

Contribution 4.2 The Hedwig approach has shown to be successful in a biomedical use case in analyzing chromosome aberrations in a 0-1 multi-resolution setting. More specifically, Hedwig was used to describe mixture-model clusters of patients described using aberration information on chromosome regions. Hedwig was also applied to describe mixture-model clusters on four non-biomedical datasets with DBpedia as background knowledge. On three out of four cases the methodology proved to be successful and we also outline when the methodology does not work.

We consider this result, published in a conference paper [27] and a journal article [24], to be one of the *main scientific contributions* of the thesis.

Contribution 4.3 The Hedwig approach was also applied in a financial domain, with the goal to analyze financial news in search for interesting vocabulary patterns from a big collection of financial articles.

The result was published in two conference papers [22], [28].

The results of this objective are described in Chapter 5.

1.4 Main Publications Related to the Thesis

Journal Articles

A. Vavpetič and N. Lavrač, "Semantic subgroup discovery systems and workflows in the SDM-toolkit," *The Computer Journal*, vol. 56, no. 3, pp. 304–320, 2013 (included in Chapter 4 of this thesis).

A. Vavpetič, V. Podpečan, and N. Lavrač, "Semantic subgroup explanations," *J. Intell. Inf. Syst.*, vol. 42, no. 2, pp. 233–254, 2014 (included in Chapter 5 of this thesis).

M. Perovšek, A. Vavpetič, J. Kranjc, B. Cestnik, and N. Lavrač, "Wordification: Propositionalization by unfolding relational data into bags of words," *Expert Syst. Appl.*, vol. 42, no. 17-18, pp. 6442–6456, 2015 (not included in this thesis).

P. R. Adhikari, A. Vavpetič, J. Kralj, N. Lavrač, and J. Hollmén, "Explaining mixture models through semantic pattern mining and banded matrix visualization," *Machine Learning Journal*, in press 2016 (included in Chapter 5 of this thesis).

Conference Papers

N. Lavrač, A. Vavpetič, L. Soldatova, I. Trajkovski, and P. Kralj Novak, "Using ontologies in semantic data mining with SEGS and g-SEGS," in *Proceedings of the International Conference on Discovery Science (DS '11)*, Springer, 2011, pp. 165–178.

M. Perovšek, A. Vavpetič, and N. Lavrač, "A wordification approach to relational data mining: Early results," in *Late Breaking Papers of the 22nd International Conference on Inductive Logic Programming, Dubrovnik, Croatia, September 17-19, 2012*, F. Riguzzi and F. Zelezný, Eds., ser. CEUR Workshop Proceedings, vol. 975, CEUR-WS.org, 2012, pp. 56–61. [Online]. Available: `http://ceur-ws.org/Vol-975`.

A. Vavpetič, V. Podpečan, S. Meganck, and N. Lavrač, "Explaining subgroups through ontologies," in *PRICAI 2012: Trends in Artificial Intelligence - 12th Pacific Rim International Conference on Artificial Intelligence, Kuching, Malaysia, September 3-7, 2012. Proceedings*, P. Anthony, M. Ishizuka, and D. Lukose, Eds., ser. Lecture Notes in Computer Science, vol. 7458, Springer, 2012, pp. 625–636, ISBN: 978-3-642-32694-3. DOI: `10.1007/978-3-642-32695-0`. [Online]. Available: `http://dx.doi.org/10.1007/978-3-642-32695-0`.

M. Perovšek, A. Vavpetič, B. Cestnik, and N. Lavrač, "A wordification approach to relational data mining," in *Discovery Science - 16th International Conference, DS 2013, Singapore, October 6-9, 2013. Proceedings*, J. Fürnkranz, E. Hüllermeier, and T. Higuchi, Eds., ser. Lecture Notes in Computer Science, vol. 8140, Springer, 2013, pp. 141–154, ISBN: 978-3-642-40896-0. DOI: `10.1007/978-3-642-40897-7`. [Online]. Available: `http://dx.doi.org/10.1007/978-3-642-40897-7`.

A. Vavpetič, P. K. Novak, and N. Lavrač, "Analysing financial vocabulary using a new semantic subgroup discovery system hedwig," in *Proceedings of the 5th Jožef Stefan International Postgraduate School Students Conference*, Ljubljana, Slovenia, 23 May 2013, pp. 219–229.

A. Vavpetič, P. K. Novak, M. Grčar, I. Mozetič, and N. Lavrač, "Semantic data mining of financial news articles," in *Proceedings of Sixteenth International Conference on Discovery Science (DS 2013)*, J. Fürnkranz, E. Hüllermeier, and T. Higuchi, Eds., ser. Lecture Notes in Computer Science, vol. 8140, Springer Berlin Heidelberg, 2013, pp. 294–307, ISBN: 978-3-642-40896-0 (included in Chapter 5 of this thesis).

P. R. Adhikari, A. Vavpetič, J. Kralj, N. Lavrač, and J. Hollmén, "Explaining mixture models through semantic pattern mining and banded matrix visualization," in *Discovery Science - 17th International Conference, DS 2014, Bled, Slovenia, October 8-10, 2014. Proceedings*, S. Džeroski, P. Panov, D. Kocev, and L. Todorovski, Eds., ser. Lecture Notes in Computer Science, vol. 8777, Springer, 2014, pp. 1–12, ISBN: 978-3-319-11811-6. DOI: `10.1007/978-3-319-11812-3`. [Online]. Available: `http://dx.doi.org/10.1007/978-3-319-11812-3`.

N. Lavrač, M. Perovšek, and A. Vavpetič, "Propositionalization online," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part III*, T. Calders, F. Esposito, E. Hüllermeier, and R. Meo, Eds., ser. Lecture Notes in Computer Science, vol. 8726, Springer, 2014 (included in Chapter 3 of this thesis), pp. 456–459, ISBN: 978-3-662-44844-1. DOI: 10.1007/978-3-662-44845-8. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-44845-8.

N. Lavrač and A. Vavpetič, "Relational and semantic data mining - invited talk," in *Logic Programming and Nonmonotonic Reasoning - 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings*, F. Calimeri, G. Ianni, and M. Truszczynski, Eds., ser. Lecture Notes in Computer Science, vol. 9345, Springer, 2015, pp. 20–31, ISBN: 978-3-319-23263-8. DOI: 10.1007/978-3-319-23264-5. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-23264-5.

1.5 Thesis Structure

The thesis is structured as follows. Following the introductory Chapter 1, Chapter 2 presents the background for the thesis, together with short descriptions of the related work. In Chapter 3 we overview the RDM task and the propositionalization technique, and present our contributions to this field. Chapter 4 presents the theoretical framework of semantic subgroup discovery, as well as presentations of systems developed in this thesis and their experimental comparisons. Next, Chapter 5 presents several applications of semantic subgroup discovery. Finally, in Chapter 6 we give our concluding remarks and present ideas for future research.

Chapter 2

Background and Related Work

This chapter presents the relevant background, together with related work in using ontologies in data mining, contrasted with our own contributions.

2.1 Background

This section presents the machine learning background, including relational data mining (RDM) and inductive logic programming (ILP), and introduces ontologies and the semantic web.

2.1.1 Rule learning

In this work we are interested in symbolic data analysis techniques, which aim at finding comprehensible patterns/models in the data. Decision tree learning [3], [4] and classification rule learning [5]–[7] are popular examples of methods, which aim at building models from class labeled data, enabling classification of yet unseen examples.

Our work is fundamentally related to classification rule learning. Following [7] we can informally introduce the problem of classification rule learning as: "Given a set of training examples, find a set of classification rules that can be used for prediction or classification of new instances." Without going into details on data and rule representation, classification rules have the following basic form:

$$\texttt{IF } f_1 \texttt{ AND } f_2 \texttt{ AND ... AND } f_L \texttt{ THEN Class = } c_i$$

which is equivalent to:

$$c_i \leftarrow f_1 \wedge f_2 \wedge \ldots \wedge f_L$$

In our work, we mostly use the following notation, which is the same as the previous, apart from using commas to denote conjunctions:

$$c_i \leftarrow f_1, f_2, \ldots, f_L$$

Each rule has a *head* and a *body*. The head contains the rule conclusion, usually a class label c_i. The body contains rule conditions $f_1, f_2, \ldots f_L$, which represent instance properties for which the rule holds. These can be simple attribute-value pairs or complex features.

Furthermore, classification rule learning systems in general find a set of rules, which altogether form a classification model useful for classifying new instances. Rules can be ordered or unordered, which affects the way new instances are classified.

In the ordered case, when a new unlabeled instance arrives, the system goes through the rules trying each one to find the first rule that "fires". This means that each conjunction in the rule body holds for the given instance. The instance is then assigned the label found in that rule's head, which is then the final classification. If no rule fires for that instance, a default rule is used—usually predicting the majority class. In the unordered case, all rules are tried and the predictions of those that fire are aggregated—for example, using voting.

In contrast to classification rule learning, descriptive rule learning focuses on learning from unlabeled data, aiming to find descriptive sets of patterns describing the data [7]. In the thesis we focus mainly on supervised descriptive rule learning [8], a task at the intersection of classification rule learning and descriptive rule learning, where the goal is to induce descriptive patterns from class labeled data. More specifically, we are interested in the *subgroup discovery* task.

2.1.2 Subgroup discovery

The task addressed in this thesis is *subgroup discovery*, a data mining task at the intersection of classification and pattern discovery. The task of subgroup discovery was defined by Klösgen [29] and Wrobel [30] 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', e.g., 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 data instances.

To give a simple example, suppose a standard subgroup discovery algorithm produces two rules for a dataset with patients (with the class *cancer*=0/1) and genes as attributes (Table 2.1).

$$R_1 : cancer = 1 \leftarrow g_A = 1 \wedge g_B = 1 \wedge g_C = 0$$
$$R_2 : cancer = 1 \leftarrow g_A = 0 \wedge g_B = 1 \wedge g_D = 1$$

Figure 2.1: Two illustrative subgroup descriptions, describing two sets of patients.

Each rule defines a subgroup of patients for which the right-hand side is true - we say that these patients are *covered* by a given rule. R_1 holds for the group of patients that have genes g_A and g_B expressed (=1), and g_C not expressed (=0). In contrast, the second group of patients has the gene g_A not expressed and genes g_B and g_D expressed. While these particular subgroups do not have patients in common, it is not uncommon for subgroups to overlap.

The left-hand side of a rule is also called the *conclusion*—in this case the rules state: if the right-hand side is true for a patient, then the patient has cancer.

In a realistic scenario, these two rules would not be accurate for every patient satisfying the rules. To quantify the *quality* of a rule there is a variety of statistics available. We list the most common statistics, together with their definitions (Table 2.1).

2.1.3 Inductive logic programming and relational data mining

It is well known from the literature on Inductive Logic Programming (ILP) [13], [14] and relational data mining (RDM) [11] that the performance of data mining methods can be

Table 2.1: Common statistics used to quantify the quality of a single rule $C \leftarrow X$ induced in the subgroup discovery process.

Statistic	Definition
Confidence	$P(C\|X)$
Lift	$\frac{P(X \wedge C)}{P(X)P(C)}$
χ^2	$\frac{N \cdot (P(X \wedge C) - P(X)P(C))}{P(X)P(\neg X)P(C)P(\neg C)}$
Weighted Relative Accuracy	$P(X) \cdot (P(C\|X) - P(C))$

significantly improved if the relations among the data objects are taken into account. In other words, the knowledge discovery process can significantly benefit from the relational domain (background) knowledge.

ILP systems use first-order logic, which provides a richer knowledge representation formalism. This allows the use of not only generally valid domain knowledge, but also the *structure* of objects that are the focus of the learning task. While the additional knowledge increases the search space, making the process much more computationally expensive, the use of background knowledge can lead to the induction of better patterns/models.

One of the most successful ILP systems is Aleph [31], which can be thought of as an ILP Swiss-army knife, since it can be used in numerous ways (feature construction, theory induction, tree induction, etc). As will become apparent in the following chapters, Aleph was incorporated into our ILP/RDM package, as well as adapted to be used for semantic subgroup discovery under the name SDM-Aleph.

One of the approaches to RDM is *propositionalization*, where the main idea is to transform a relational problem (with certain properties) into a propositional problem, which can then be solved using traditional propositional learners. This transformation is useful, since it can be used with any data mining or machine learning task in mind, as well as with a plethora of readily available learners. RDM and propositionalization are discussed in more detail in Chapter 3.

2.1.4 Semantic web and ontologies

A special form of background knowledge, which has not been exploited in the original ILP and RDM literature, are ontologies. The concept of "ontology" comes from philosophy. Hofweber neatly expresses the main two points of the philosophical ontology as: "first, say what there is, what exists, what the stuff in reality is made out of, secondly, say what the most general features and relations of these things are" [32].

Similarly, in computer science and artificial intelligence, we wish to design a representational artifact that is intended to represent entities and relations among them in one domain or across several, with the main goal of enabling easier understanding and cooperation between distinct information systems. Smith [33] defines an ontology in the context of information science as follows: "An ontology is in this context a dictionary of terms formulated in a canonical syntax and with commonly accepted definitions designed to yield a lexical or taxonomical framework for knowledge-representation which can be shared by different information systems communities."

While in philosophy "ontology" exists only in its singular form, we tend to use "ontologies" also in plural. This is to emphasize the fact that they are *separate* domain models that come from different sources, even though the models can always be joined under a common root concept into one model.

A domain ontology can also act as a mean of providing additional information to machine learning (data mining) algorithms by attaching semantic descriptors to the data. Such domain knowledge is usually represented in a standard format which 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 tool for building ontologies).

The OWL family of languages is based on *Description Logics* (DL). DL are a family of knowledge representation languages, many of which are more expressive than propositional logic but less expressive than first-order logic (FOL). Due to this many DL languages have decidable reasoning problems solvable with efficient algorithms [34]. There are notions that are equal between FOL, DL and OWL but have different names in their respective communities. To minimize confusion, we list the synonyms of several notions in Table 2.2.

Table 2.2: Synonyms of notions equivalent across FOL, OWL and DL. Taken from Wikipedia (`https://en.wikipedia.org/wiki/Description_logic`) on January 14, 2016.

FOL	OWL	DL
unary predicate	class	concept
binary predicate	property	role
constant	individual	individual

As mentioned above, DL are a family of languages with varying expressiveness. The expressiveness of a language is typically encoded in a label, such as $\mathcal{SHOIN}^{(\mathcal{D})}$ (which is equivalent to OWL DL), $\mathcal{EL}$ or $\mathcal{ALC}$, where the letters of the label encode the properties of the language.

For example, the meaning of $\mathcal{ALC}$ can be read as follows. The prefix $\mathcal{AL}$ denotes that this is an attributive language, supporting atomic negation, concept intersection, universal restrictions, and limited existential quantification. The $\mathcal{C}$ denotes that the language also supports complex concept negation. For a complete overview of the DL naming nomenclature refer to [35].

On the other hand, the RDF data model is simple, yet powerful, and compatible with OWL. 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. The Linking Open Data project[5] aims at publishing

[1] `http://www.w3.org/OWL/`
[2] `http://www.w3.org/RDF/`
[3] `http://protege.stanford.edu/`
[4] `http://linkeddata.org/`
[5] https://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

and connecting openly available datasets in RDF, and the number of added datasets has grown substantially (see Figure 2.2; credit for the diagrams goes to Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak of `http://lod-cloud.net/`).

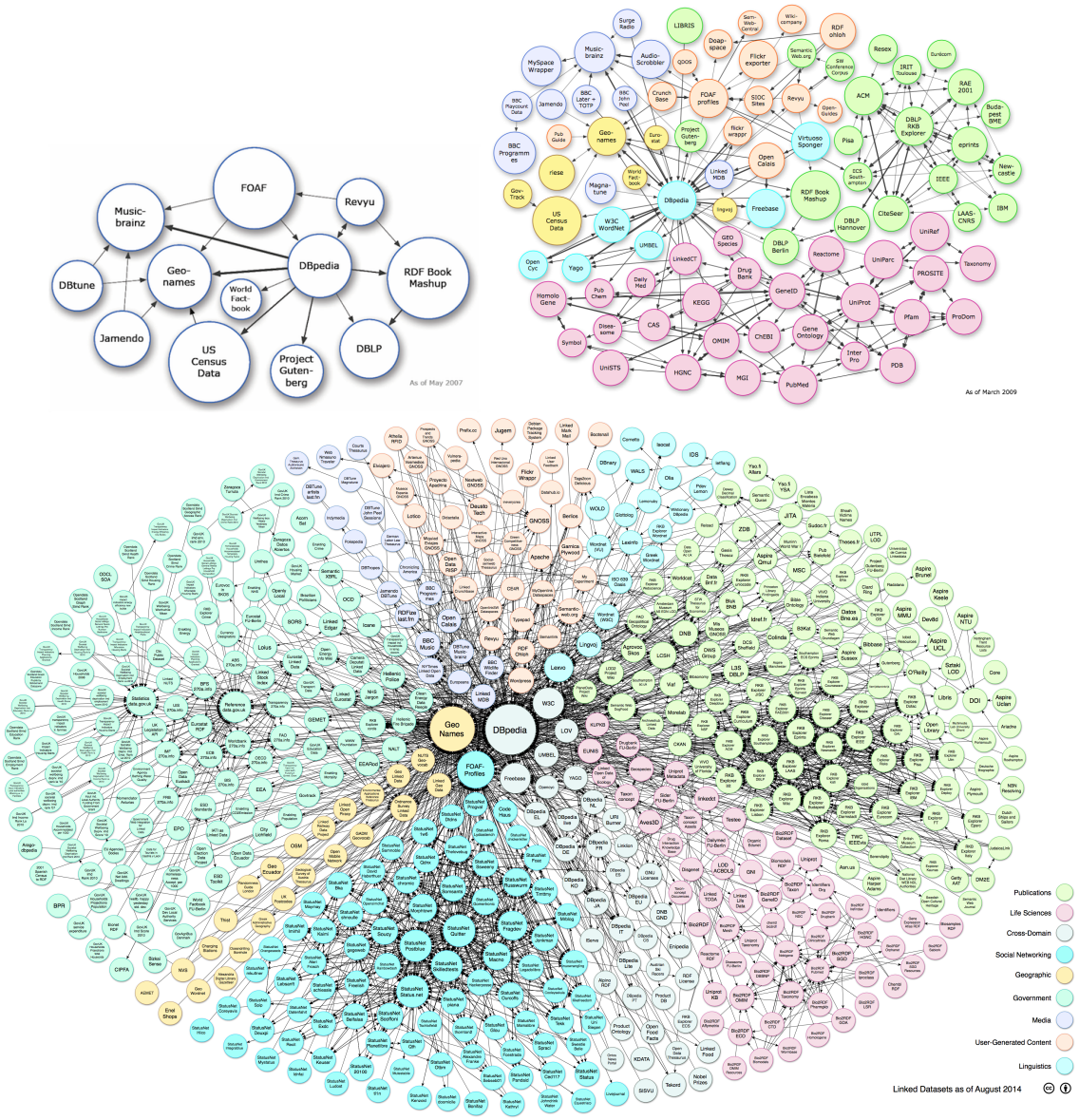


Figure 2.2: The Linked Open Data graphs from 2007, 2009 and 2014, showing a substantial growth: from 12, 89 to 570 datasets.

2.2 Related Work

This thesis is mostly concerned with semantic subgroup discovery, i.e., using domain knowledge encoded as ontologies to improve the process of subgroup discovery.

A related task of mining association rules with taxonomies has been studied in [36]–[38], where taxonomies can be essentially thought of as ontologies with only one type of relation: *is-a*. In this task, items are part of a taxonomy (a directed acyclic graph, or DAG), and associations between items can be detected at any level of the taxonomy. Using

taxonomies to speed up propositionalization, as well as the subsequent step of classification rule learning using a feature generality taxonomy, is proposed in [39].

In [40], background knowledge is encoded in the inheritance network notation and the developed KBRL algorithm performs a general-to-specific heuristic search for a set of conjunctive rules that satisfy user-defined rule evaluation criteria. In [41], 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 anti-chain from a taxonomy that improves the classification accuracy.

In [42] an engineering ontology of CAD (Computer-Aided Design) 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 [43]. 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.

Kietz [44] was one of the first to make a step in the direction of adapting existing relational learners to use DL by extending the standard learning bias used in ILP with DL (CARIN-$\mathcal{ALN}$). More recently, Lehmann and Haase [45] defined a refinement operator in the $\mathcal{EL}$ DL; they consider the construction of consistent and complete hypotheses using an ideal refinement operator. In [46], 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 [47]. The initial work in that direction includes [48]–[50], 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.

In this thesis we focus on the problem of semantic subgroup discovery, which has not been addressed in the related work so far. One crucial step in this direction was made in [18] with the system SEGS (Search for Enriched Gene Sets) - a domain specific system that uses ontologies and other hierarchies as background knowledge for data mining. SEGS upgrades previous approaches to gene set enrichment analysis [51], [52]. Compared to earlier work in gene set enrichment [51], [52], the novelty of SEGS is that it does not only 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. In this thesis we take this idea further by generalizing it to be applicable in any domain, as well as employing more expressive input and hypothesis languages.

Chapter 3

Relational Data Mining Framework

This chapter presents our contributions to the relational data mining (RDM) field. First, we define the RDM task. Second, we explain the *propositionalization* technique, together with a brief summary of our experimental results on the performance of several such techniques. We also present our approach to making the use of propositionalization algorithms easier for non-experts, as well as making experiments shareable and repeatable.

3.1 Relational Data Mining Task Formulation

Standard machine learning and data mining algorithms induce hypotheses in the form of models or propositional patterns learned from a given data table, where one example corresponds to a single row in the table. Most types of propositional models and patterns have corresponding relational counterparts, such as relational classification rules, relational regression trees, relational association rules. Inductive Logic Programming (ILP) [13] and Relational Data Mining (RDM) [11], [14] algorithms can be used to induce such relational models and patterns from multi-relational data, e.g., data stored in a relational database.

In terms of database terminology, a relational data mining task can be formally defined as follows.

Given:

- a target table t, where each row is one example,
- related tables T, connected to t via foreign keys.

Find:

- a query Q (a set of sub-queries) that together with T describes the target properties of t.

where the target property can be a selected class label (the target attribute value) or some other property of interest.

Table 3.1 shows an simple relational database schema for storing authorship information about researchers, their fields of research, and papers. Suppose we are interested in finding

Table 3.1: An example relational database schema. Underlined attributes denote the private and foreign keys connecting the tables.

researcher		
id	name	field

author	
researcherId	paperId

paper		
id	title	conference

descriptions of researchers from the field of data mining. An example result (in Prolog-esque syntax) of an RDM algorithm could be something like:

```
researcher(R, _, 'Data Mining') ← author(R, P), paper(P, _, 'ECML/PKDD')
```

The body of this rule yields researchers that have published papers at the ECML/PKDD conference, which the algorithm determined as a good pattern for describing data mining researchers. The rule references the *author*, *paper* and the main *researcher* table via foreign keys. In this way the algorithm exploited the *structural* information available for the learning examples (i.e., researchers), which illustrates the main powerful feature of RDM.

RDM problems are characterized by multiple relations and can be tackled in two different ways: (1) by using a relational learner such as Progol [53] or Aleph [31], which can build a model or induce a set of patterns directly, or (2) by constructing complex relational features used to transform the relational representation into a propositional format and then applying a propositional learner on the transformed single-table representation. The latter approach, called *propositionalization*, is described in detail in the next section.

3.2 Propositionalization

For relational databases in which data instances are clearly identifiable (the so-called *individual-centered representation* [54]), various techniques can be used for transforming a relational database into a propositional single-table representation [55]. After performing such a transformation [56], typically called *propositionalization* [57], standard propositional learners can be used, including decision tree and classification rule learners.

Propositional representations (a single table format) impose the constraint that each training example is represented as a single fixed-length tuple. Due to the nature of some relational problems, there exists no propositional encoding; for example, the authorship network introduced in the previous section in general cannot be represented in a propositional format without loss of information, since each researcher can have any number of co-authors and papers. The problem is naturally represented using multiple relations, i.e., using the *researcher*, *author* and *paper* relations.

Propositionalization is a form of *constructive induction*, since it involves changing the representation for learning. As we noted before, propositionalization cannot always be done without loss of information, but it can be a powerful method when a suitable relational learner is not available or when a non-conventional ILP task needs to be performed on data from a given relational database (e.g., clustering). As mentioned, the problem at hand must be *individual-centered* [54]. Such problems have a clear notion of an individual and the learning occurs only at the level of (sets of) individual instances rather than the (network of) relationships between the instances. As an example, consider the problem of classifying researchers into research fields given a co-authorship network; in this case the researcher is an individual and learning occurs at the researcher's level, i.e. assigning class labels to researchers.

To illustrate the propositionalization scenario, consider a simplified multi-relational problem, where the data to be mined is a database of authors and their papers, with the task of assigning a research field to unseen authors. In essence, a complete propositional representation of the problem (shown in Table 3.2 would be a set of queries $q \in Q$ (complex relational features) that return value *true* or *false* for a given author. Each query describes a property of a researcher. The property can involve a rather complex query, involving multiple relations as long as that query returns either true or false, or the result of some other aggregation function. For example, a query could be "does author X have a paper

Table 3.2: A sample propositional representation of the *researcher* table.

researcher					
id	q_1	q_2	...	q_m	field
R_1	1	1	...	1	F_1
R_2	0	1	...	0	F_1
R_3	1	0	...	0	F_2
...	...	...	...	...	...
R_n	0	1	0	0	F_1

published at the ECML/PKDD conference?" or "how many papers does author X have published at the ECML/PKDD conference?".

While this transformation could be done manually by a data analyst, we are only interested in automated propositionalization methods. Furthermore, the transformation into a propositional representation can be done with essentially any ML or DM task in mind: classification, association discovery, clustering, etc.

3.3 Implementantion of Selected RDM Techniques

ILP and RDM approaches are flexible tools, which can also be effectively exploited for semantic data mining (as shown in our own research; see Chapter 4). During our study of ILP & RDM topics, it became apparent that these techniques are far from accessible (especially ILP approaches) even for data mining researchers. One of the main reasons is that each approach enforces its own input format and hypothesis language definitions, with poor support for actually working with data stored in relational databases.

For this reason, we developed the open-source `python-rdm` library and a number of widgets in an open-source web-based data mining platform ClowdFlows [58]. Using the Python library, researchers are able to easily include RDM techniques into their own experiments, while the ClowdFlows widgets allow simple sharing and repeatability of experiments. Both aim to alleviate much of the issues a researcher new to ILP and RDM would otherwise encounter.

3.3.1 Python relational data mining library

Our python library, dubbed `python-rdm`, provides a common interface to several algorithms[1], including the popular ILP system Aleph [31] together with its feature construction component, as well as 1BC [59] and 1BC2 [60] first-order Bayesian classifiers, and the Tertius [61] first-order rule learner. It also provides RSD [62], RelF [63], Relaggs [64], Quantiles and Cardinalization [65], and Wordification [19] propositionalization approaches. Our software offers support for working with data stored in MySQL and PostgreSQL databases.

Our ClowdFlows package, described in the following paragraphs, internally uses the `python-rdm` library. Likewise, we envision that the library can be used by researchers to easily prototype new solutions or to include relational data mining approaches in their experiments.

[1] 1BC, 1BC2, Tertius, Relaggs, Quantiles, Cardinalization and PostgreSQL support were added in collaboration with Nicolas Lachiche and Alain Shakour, University of Strasbourg.

3.3.2 ClowdFlows relational data mining package

The ClowdFlows platform [58] is an open-source, web-based data mining platform that supports the construction and execution of scientific workflows. This web application can be accessed and controlled from anywhere while the processing is performed in a cloud of computing nodes. A public installation of ClowdFlows is accessible at `http://clowdflows.org`. For a developer, the graphical user interface supports simple operations that enable workflow construction: adding workflow components (widgets) on a canvas and creating connections between the components to form an executable workflow, which can be shared by other users or developers. Upon registration, the user can access, execute, modify, and store the modified workflows, enabling their sharing and reuse. On the other hand, by using anonymous login, the user can execute a predefined workflow, while any workflow modifications would be lost upon logout.

We have extended ClowdFlows with the implementation of an ILP/RDM toolkit, including support for MySQL and PostgreSQL databases and a variety of algorithms listed in the previous section. The construction of RDM workflows is supported by other specialized RDM components (e.g., the MySQL package providing access to a relational database by connecting to a MySQL database server), other data mining components (e.g., the Weka [66] classifiers) and other supporting components (including cross-validation), accessible from other ClowdFlows modules. Each public workflow is assigned a unique URL that can be accessed by any user to either repeat the experiment, or use the workflow as a template to design another workflow. Consequently, the incorporated RDM algorithms become handy to use in real-life data analytics, which may therefore contribute to improved accessibility and popularity of ILP and RDM.

In terms of workflows reusability, accessible by a single click on a web page where a workflow is exposed, the implemented propositionalization toolkit is a significant step towards making the ILP legacy accessible to the research community in a systematic and user-friendly way. To the best of our knowledge, this is the only workflow-based implementation of ILP and RDM algorithms in a platform accessible through a web browser, enabling simple workflow adaptation to the user's needs.

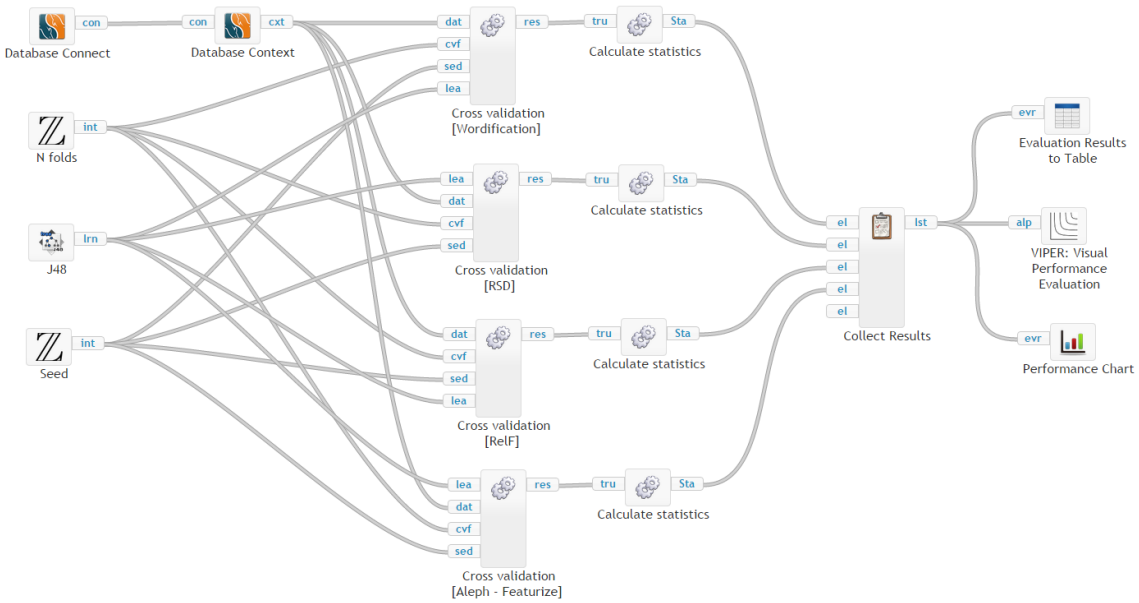


Figure 3.1: Evaluation workflow for evaluating and comparing Wordification, Aleph, RSD, and RelF, implemented in the ClowdFlows data mining platform.

There are several ready-to-use workflows available within the public installation of ClowdFlows[2]. Here we list a few selected workflows:

- A very simple workflow that reads the examples and background knowledge as Prolog facts, and uses RSD together with J48 to construct a decision tree[3].
- A workflow that reads data from a MySQL database, constructs features using RSD and finally constructs and visualizes a J48 decision tree[4],
- Workflow similar to the previous one, except for using Aleph as a propositionalization tool[5],
- Evaluation and visualization workflow, comparing four propositionalization approaches[6]; see Figure 3.1.

3.4 Software Availability

The `python-rdm` package is open-source and available on GitHub[7]. The repository contains the Python library, the ClowdFlows package, unit tests, and documentation. The authors welcome extensions and improvements from the community.

The ClowdFlows platform, mainly developed by Janez Kranjc, is open-source and available on GitHub[8], together with instructions to host your own instance of ClowdFlows. However, a public installation is available at `http://clowdflows.com`, together with our ILP/RDM widgets.

3.5 Related Publication

Some of the ClowdFlows implementations of relational data mining are described in the following publication (included in this chapter):

N. Lavrač, M. Perovšek, and A. Vavpetič, "Propositionalization online," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part III*, T. Calders, F. Esposito, E. Hüllermeier, and R. Meo, Eds., ser. Lecture Notes in Computer Science, vol. 8726, Springer, 2014, pp. 456–459, ISBN: 978-3-662-44844-1. DOI: `10.1007/978-3-662-44845-8`. [Online]. Available: `http://dx.doi.org/10.1007/978-3-662-44845-8`.

The authors' contributions are as follows. Nada Lavrač was the leader of the project and the main author of the text. Matic Perovšek designed the included experiments, while Anže Vavpetič implemented the ILP and MySQL packages described in the paper. All authors contributed to the text of the publication.

[2] http://clowdflows.com/existing-workflows/
[3] http://clowdflows.org/workflow/471/
[4] http://clowdflows.org/workflow/611/
[5] http://clowdflows.org/workflow/2224/
[6] http://clowdflows.org/workflow/4018/
[7] `https://github.com/xflows/rdm`
[8] https://github.com/xflows/clowdflows

Propositionalization Online

Nada Lavrač[1,2,3], Matic Perovšek[1,2], and Anže Vavpetič[1,2]

[1] Jožef Stefan Institute, Ljubljana, Slovenia
[2] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
[3] University of Nova Gorica, Nova Gorica, Slovenia
{nada.lavrac,matic.perovsek,anze.vavpetic}@ijs.si

Abstract. Inductive Logic Programming and Relational Data Mining address the task of inducing models or patterns from multi-relational data. An established relational data mining approach is propositionalization, characterized by transforming a relational database into a single-table representation. The paper presents a propositionalization toolkit implemented in the web-based data mining platform ClowdFlows. As a contemporary integration platform it enables workflow construction and execution, provides open access to Aleph, RSD, RelF and Wordification feature construction engines, and enables RDM performance comparison through cross-validation and ViperCharts results visualization.

Keywords: relational data mining, propositionalization, web access.

1 Introduction

Propositional data mining algorithms induce hypotheses in the form of models or patterns learned from a given data table. In contrast, Inductive Logic Programming (ILP) [6] and Relational Data Mining (RDM) [1] algorithms induce models or patterns from multi-relational data (e.g., relational databases). For relational databases with clearly identifiable instances (i.e., *individual-centered representations* [2], characterized by one-to-many relationships among data tables), propositionalization techniques [3] can be used to transform a relational database into a propositional single-table format, followed by propositional learning, e.g., by using a decision tree or a classification rule learner.

This paper presents an online propositionalization toolkit, which can be used to construct RDM workflows. As completed workflows, data, and results can be made public by the author of the workflow, the platform can serve as an easy-to-access integration platform for various RDM workflows.

2 Clowdflows ILP module

The ClowdFlows platform [4] is an open-source, web-based data mining platform that supports the construction and execution of scientific workflows. This web application can be accessed and controlled from anywhere while the processing is performed in a cloud of computing nodes. A public installation of ClowdFlows

T. Calders et al. (Eds.): ECML PKDD 2014, Part III, LNCS 8726, pp. 456–459, 2014.

is accessible at `http://clowdflows.org`. For a developer, the graphical user interface supports simple operations that enable workflow construction: adding workflow components (widgets) on a canvas and creating connections between the components to form an executable workflow, which can be shared by other users or developers. Upon registration, the user can access, execute, modify, and store the modified workflows, enabling their sharing and reuse. On the other hand, by using anonymous login, the user can execute a predefined workflow, while any workflow modifications would be lost upon logout.

We have extended ClowdFlows with the implementation of an ILP toolkit, including the popular ILP system Aleph [9] together with its feature construction component, as well as RSD [10], RelF [5] and Wordification [7] propositionalization engines. Construction of RDM workflows is supported by other specialized RDM components (e.g., the MySQL package providing access to a relational database by connecting to a MySQL database server), other data mining components (e.g., the Weka classifiers) and other supporting components (including cross-validation), accessible from other ClowdFlows modules. Each public workflow is assigned a unique URL that can be accessed by any user to either repeat the experiment, or use the workflow as a template to design another workflow. Consequently, the incorporated RDM algorithms become handy to use in real-life data analytics, which may therefore contribute to improved accessibility and popularity of ILP and RDM.

Figure 1 shows two simple workflows using the ILP and Weka module components. The first workflow assumes that the user uploads the files required by RSD

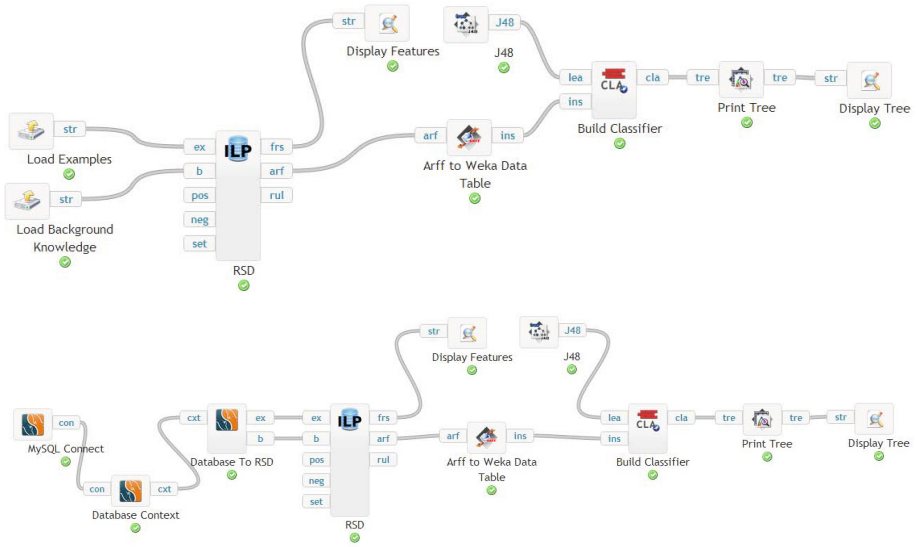


Fig. 1. Above: Simple RSD propositionalization workflow using ILP and Weka components, available online at `http://clowdflows.org/workflow/471/`. Below: The same RSD workflow, extended by accessing the training data using a MySQL database, available at `http://clowdflows.org/workflow/611/`.

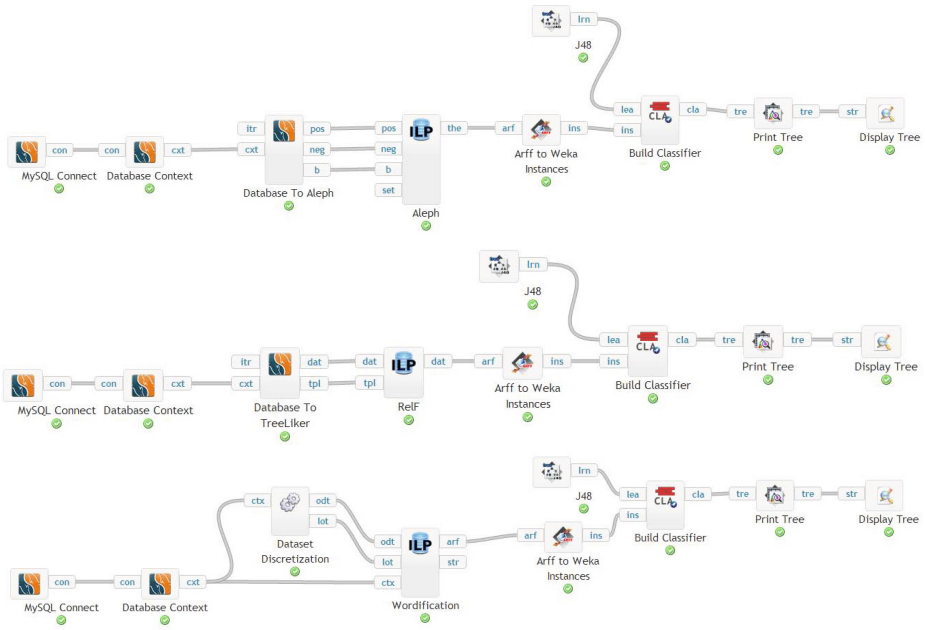


Fig. 2. Propositionalization workflows available online: for Aleph at `http://clowdflows.org/workflow/2224/`, for RelF at `http://clowdflows.org/workflow/2227/` and for Wordification at `http://clowdflows.org/workflow/2222/`.

as Prolog programs, while the second workflow extends this use case by retrieving the training data from a MySQL database server and automatically constructing the background knowledge and the training examples. Similar workflows, constructed for the other three propositionalization approaches Aleph, RelF and Wordification, are illustrated in Figure 2.

The evaluation workflow is shown in Figure 3. After reading the relational data and data discretization, propositionalization algorithms are applied, their results are transformed into the Weka input format for the J48 decision tree learner, followed by 10-fold cross-validation with identical folds allowing performance comparison of different propositionalization algorithms. The results of cross-validation (precision, recall, F-score) are connected to the input of VIPER (Visual Performance Evaluation) engine [8], which displays the results as points in the precision-recall space. The evaluation workflow enables ILP researchers to reuse the developed workflow and its components in future experimentation.

In terms of workflows reusability, accessible by a single click on a web page where a workflow is exposed, the implemented propositionalization toolkit is a significant step towards making the ILP legacy accessible to the research community in a systematic and user-friendly way. To the best of our knowledge, this is the only workflow-based implementation of ILP and RDM algorithms in a platform accessible through a web browser, enabling simple workflow adaptation to the user's needs.

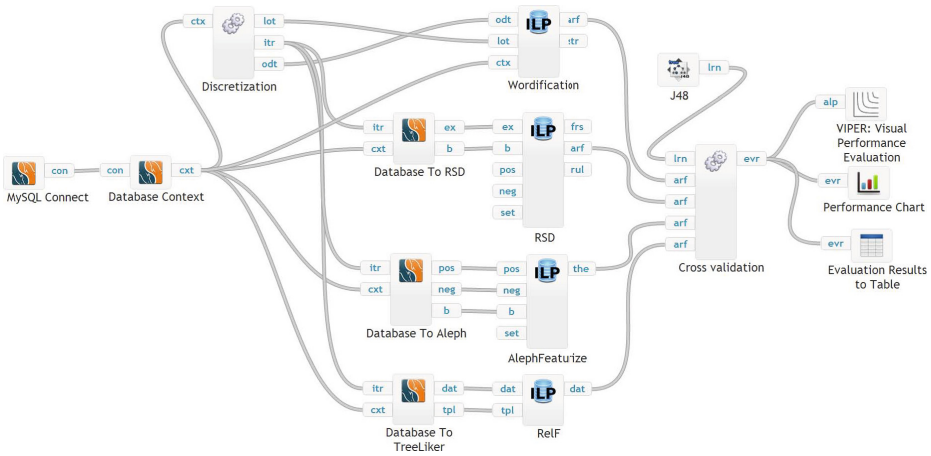


Fig. 3. Performance evaluation workflow, available at `http://clowdflows.org/workflow/2210/`, comparing the results of J48 after propositionalization by Aleph, RSD, RelF and Wordification.

References

[1] Džeroski, S., Lavrač, N. (eds.): Relational Data Mining. Springer (2001)

[2] Flach, P.A., Lachiche, N.: 1BC: A First-Order Bayesian Classifier. In: Džeroski, S., Flach, P.A. (eds.) ILP 1999. LNCS (LNAI), vol. 1634, pp. 92–103. Springer, Heidelberg (1999)

[3] Kramer, S., Lavrač, N., Flach, P.A.: Propositionalization approaches to relational data mining. In: Džeroski and Lavrač pp. 262–292

[4] Kranjc, J., Podpečan, V., Lavrač, N.: ClowdFlows: A cloud based scientific workflow platform. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) ECML PKDD 2012, Part II. LNCS, vol. 7524, pp. 816–819. Springer, Heidelberg (2012)

[5] Kuželka, O., Železný, F.: Block-wise construction of tree-like relational features with monotone reducibility and redundancy. Machine Learning 83(2), 163–192 (2011)

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

[7] Perovšek, M., Vavpetič, A., Cestnik, B., Lavrač, N.: A wordification approach to relational data mining. In: Fürnkranz, J., Hüllermeier, E., Higuchi, T. (eds.) DS 2013. LNCS, vol. 8140, pp. 141–154. Springer, Heidelberg (2013)

[8] Sluban, B., Lavrač, N.: ViperCharts: Visual performance evaluation platform. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) ECML PKDD 2013, Part III. LNCS, vol. 8190, pp. 650–653. Springer, Heidelberg (2013)

[9] Srinivasan, A.: Aleph manual (March 2007), `http://www.cs.ox.ac.uk/activities/machinelearning/Aleph/`

[10] Železný, F., Lavrač, N.: Propositionalization-based relational subgroup discovery with RSD. Machine Learning 62(1-2), 33–63 (2006)

Chapter 4

Semantic Subgroup Discovery

In this chapter we first present a motivating example, followed by introducing a theoretical framework of semantic subgroup discovery (SSD). Next, we present the systems developed as part of this thesis that solve the SSD task: SDM-SEGS, SDM-Aleph and Hedwig. Lastly, we present some experimental comparisons.

4.1 Motivating Example

As a motivating example of semantic data mining [16], consider a bank which has the following data about its clients: occupation, place of living, bank services used, which includes the account type, possible credits, insurance policies, etc. The bank also categorized the clients as 'big spenders' or not and wants to find patterns describing the former. Table 4.1 presents the input data—the *client* relation, as well as additional background or structural information, i.e. the *married* relation.

Suppose we also have three ontologies: an ontology of banking services, an ontology of locations and an ontology of occupations, shown in Figure 4.1.

Table 4.1: Table of bank clients described by different attributes and class 'big spender', and the relational table connecting different clients that are married.

client					
id	occupation	location	account	...	big spender
0	Doctor	Munich	Gold	...	yes
1	Nurse	Rome	Classic	...	yes
2	Finance	Krakow	Gold	...	yes
...	...	...	...		
27	Retail	Bologna	Classic	...	no
28	Finance	Nuremberg	Classic	...	no
29	Nurse	Augsburg	Student	...	no

married	
client1Id	client2Id
0	11
1	2
4	17
...	...

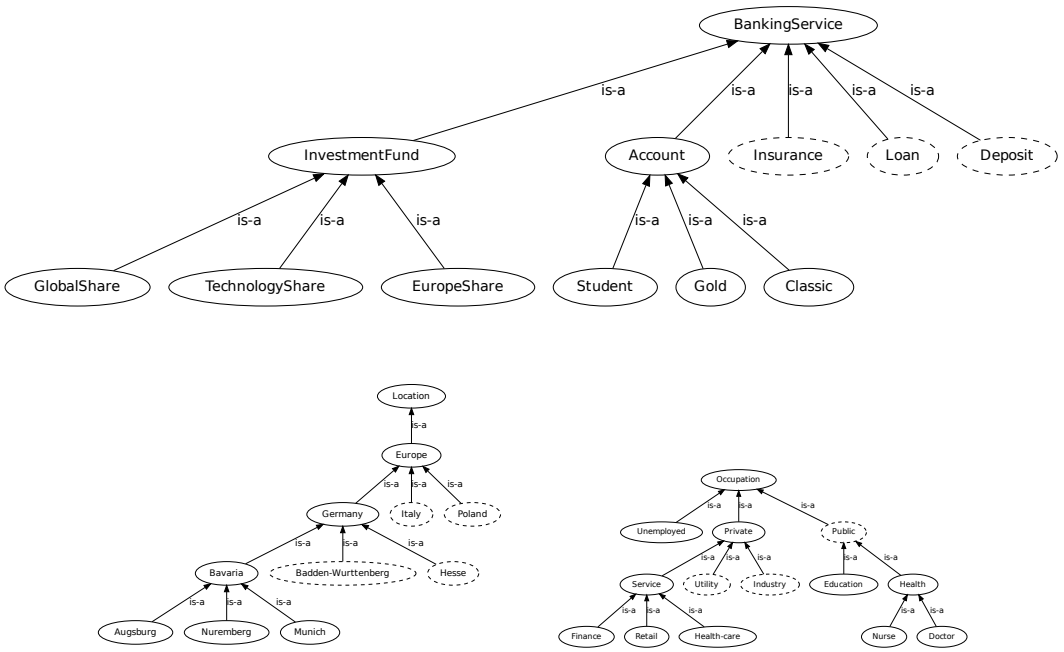


Figure 4.1: The ontologies of banking services, locations and occupations. Concepts with omitted sub-concepts are drawn with a dashed line.

```
big_spender(X) ← married(X, Y),
                 has_occupation(Y, healthSector),
                 uses_service(Y, goldAcc)

big_spender(X) ← has_occupation(X, doctor),
                 uses_service(X, deposit)

big_spender(X) ← lives_in(X, germany),
                 has_occupation(X, serviceSector),
                 uses_service(X, investment_fund)
```

Figure 4.2: Three subgroup descriptions discovered in the banking domain. Each subgroup description represents a group of big spenders.

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

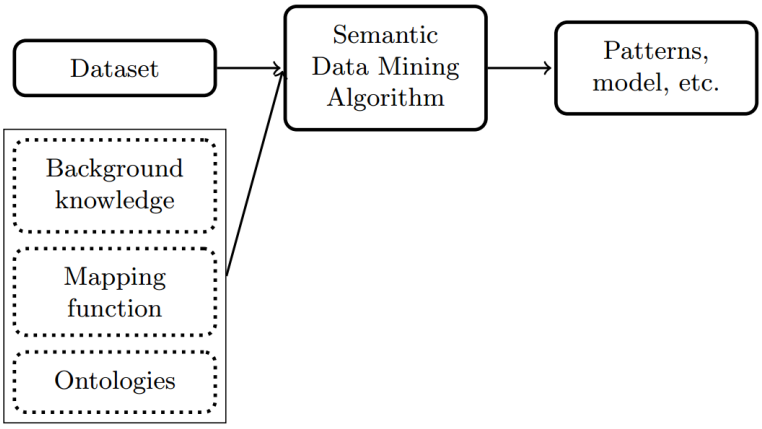


Figure 4.3: The Semantic Data Mining (SDM) process illustration.

in the data itself. A possible pattern in this domain could be e.g., `big_spender(X)` ← `lives_in(X, germany)`, describing all examples/people living in Germany, although in the data table we only have the information on specific German cities.

Figure 4.2 presents a subset of subgroup descriptions discovered on the banking domain.

4.2 Semantic Subgroup Discovery Problem Definition

In this thesis we focus on a particular semantic data mining task, i.e., semantic subgroup discovery. In this section we formally define the task problem. We define semantic subgroup discovery by extending the (relational) subgroup discovery problem definition. We start using the *semantic data mining definition* from [17] (see also Figure 4.3):

Given:

- *domain knowledge* in the form of ontologies,
- a set of *training examples* (experimental data),
- *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.

We make the definition more specific to get the *semantic subgroup discovery task definition*:

Given:

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

Find: *population subgroups* that are statistically 'most interesting', e.g., are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest [29], [30].

Next, we present our implementation of the above definition more formally. Namely, various objectives functions can be found in subgroup discovery literature, all of which can be argued find subgroups with the properties described in the task definition. Frequently the objective is to find the top-k subgroups, i.e. the algorithm should list k subgroups with the best score according to some rule quality function. A common problem of this sort of algorithms is that such rules can often be very similar—they cover approximately the same group of examples and are thus not as useful to the domain expert. In this work we add an additional component to the objective function, which accounts also for too similar rules.

Given:

- data description language $\mathcal{L}$, imposing a bias on the form of data,
- a hypothesis description language $\mathcal{H}$, imposing a bias on the form of rules,
- a set of class-labeled training examples $\mathcal{E}$,
- background knowledge $\mathcal{B}$, which defines the relations that are not part of the ontology,
- a domain ontology $\mathcal{O} =< C, \leq_C, R, A >$, comprised of concepts C, the *generalization* structure $\leq_C$, relation identifiers R, and a set of axioms A,
- a mapping function $M : \mathcal{E} \rightarrow 2^C$,
- a rule quality function ϕ,
- user-defined number of rules k.

Find: a set of rules $\mathcal{R}$ described in the hypothesis description language, such that

$$\arg\max_{\mathcal{R}} \frac{\sum_{r \in \mathcal{R}} \phi(r)}{\sum_{r_i, r_j \in \mathcal{R}, i \neq j} |r_i \cap r_j| + 1}$$

where $|\mathcal{R}| = k$.

Essentially, we want to maximize rule quality of the set of rules (the numerator), while at the same time having the rules cover different parts of the example space (the denominator).

Note that for $\mathcal{B} = \emptyset, \mathcal{O} = \emptyset$ we have the propositional subgroup discovery task and for $\mathcal{O} = \emptyset$ we have relational subgroup discovery. The key element of semantic subgroup discovery therefore lies in $\mathcal{O}$, which contains *domain meta information*, i.e., information about attributes describing the training examples.

Furthermore, by selecting a suitable hypothesis language, the framework also encompasses supervised tasks—namely the supervised descriptive rule induction tasks (subgroup discovery, contrast set mining, etc).

4.3 Semantic Subgroup Discovery Algorithms SDM-SEGS and SDM-Aleph

This section presents our own semantic subgroup discovery SDM-SEGS and SDM-Aleph and provides a brief overview of the journal paper. The full paper is included at the end of the chapter.

4.3.1 SDM-SEGS

This section describes the semantic subgroup discovery system SDM-SEGS which 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 similar manner 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, SDM-SEGS 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 our system SDM-SEGS: the input data, the hypothesis language, the rule construction algorithm and the rule selection and evaluation principles.

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[1],
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.

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 which represent ontological concepts or a binary relation between examples.

Rule construction A set of rules which satisfy the size constraints (minimum support and maximum number of rule terms) is constructed using a top-down bounded *exhaustive* search algorithm, which enumerates all the possible rules by taking exactly one term from each (sub-)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 which would have been generated if this rule were further specialized.

[1] Unlike SDM-Aleph described in Section 4.3.2, SDM-SEGS exploits only the *concept* and *subClassOf* relations.

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 which selects the best rules according to the *wWRAcc* (Weighted Relative Accuracy with example weights) heuristic [67]. The weighted covering algorithm uses example weights as means for considering different parts of the example space when selecting the best rules.

4.3.2 SDM-Aleph

In this section we present the semantic subgroup discovery system SDM-Aleph, based on the ILP system Aleph[2]. 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 the system, its hypothesis language and the used rule construction and selection techniques.

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[3]),
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.

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.

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[4]:

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

[3] 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.

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

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

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 [53].

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.

4.4 Semantic Subgroup Discovery with Hedwig

This section presents our latest semantic subgroup discovery system Hedwig [28]. We provide a detailed description, together with an experimental comparison to previous approaches.

4.4.1 Hedwig algorithm

Compared to standard subgroup discovery algorithms, Hedwig uses domain ontologies to structure the search space and formulate generalized hypotheses [28]. Existing semantic subgroup discovery algorithms are either specialized for a specific domain [18] or adapted from systems that do not take into the account the hierarchical structure of background knowledge [17]. Hedwig overcomes these limitations as it is designed to be a general purpose semantic subgroup discovery system.

Semantic subgroup discovery, as addressed by the Hedwig system, results in relational descriptive rules. Hedwig uses ontologies as background knowledge and training examples in the form of Resource Description Framework (RDF) triples. We present the algorithm pseudo-code in Figures 4.1 and 4.2.

The Hedwig system supports ontologies and examples to be loaded as a collection of RDF triples (a graph). The system automatically parses the RDF graph for the `subClassOf` hierarchy, as well as any other user-defined (so-called) *generalization* predicates or binary relations between examples. Hedwig also defines a namespace of classes and relations for specifying the training examples to which the input must adhere.

The algorithm uses beam search, where the beam contains the best N rules found so far. The search starts with the default rule which covers all the input examples. In every iteration of the search, each rule from the beam is specialized using one of the four operations:

Algorithm 4.1: Hedwig's `induce`(E, B, c, k, α) procedure.

Input : Input examples E, background knowledge B, target class value c, beam size k, p-value threshold α
Output: Set of rules

```
rules ← [default_rule(E, c, B)]
while improvement(rules) do
    // Add specializations of each rule to the beam
    for rule ∈ rules do
        extend(rules, specialize(rule, B))
    end
    rules ← best(rules, k) // Select the top k rules
end
rules ← validate(rules, α) // Significance testing
return rules
```

Algorithm 4.2: Hedwig's `specialize`($rule$, B) procedure.

Input : Rule to specialize $rule$, background knowledge B
Output: Set of specializations of $rule$

```
specializations ← []
// Predicates that can be specialized
eligible_preds ← eligible(predicates(rule))
for predicate ∈ eligible_preds do
    // Specialize by traversing the subClassOf hierarchy
    for subclass ∈ subclasses(predicate, B) do
        new_rule ← swap(rule, predicate, subclass)
        if can_specialize(new_rule) then
            append(specializations, new_rule)
        end
    end
    // Specialize by negating
    new_rule ← negate(rule, predicate)
    if can_specialize(new_rule) then
        append(specializations, new_rule)
    end
end
if rule ≠ default_rule then
    // Specialize by adding a new unary predicate
    new_predicate ← next_non_ancestor(eligible_preds)
    new_rule ← append(rule, new_predicate)
    if can_specialize(new_rule) and non_redundant(new_rule) then
        append(specializations, new_rule)
    end
end
if is_unary(last(predicates(rule))) then
    // Specialize by adding new binary predicates
    extend(specializations, specialize_binary(new_rule))
end
return specializations
```

Table 4.2: Statistical rankings of algorithms. A is shorthand for the ALL dataset and h is shorthand for the hMSC dataset.

Measure	AUC		Coverage		Significance		Support		WRAcc		# Wins
Alg \ Data	A	h	A	h	A	h	A	h	A	h	
SDM-Aleph	1	2	1	2	2	2	1	1	2	2	4
SDM-SEGS	1	2	2	2	1	2	2	2	1	2	3
SEGS	2	2	2	2	1	1	2	2	2	1	3
Hedwig	2	**1**	**1**	**1**	2	2	2	2	**1**	**1**	**5**

1. replace predicate of a rule with a predicate that is a sub-class of the previous one,
2. negate predicate of a rule,
3. append a new unary predicate to the rule,
4. append a new binary predicate, thus introducing a new existentially quantified variable.[5]

Rule induction via specializations is a well-established way of inducing rules, since every specialization either maintains or reduces the current number of covered examples. A rule will not be specialized once its coverage is zero or falls below some predetermined threshold (e.g., minimum support). If the extended rule does not improve the probability of the conclusion (we use the redundancy coefficient, as in [68]), then it is not added to the pool of specializations. After the specialization step is applied to each rule in the beam, we select a new set of the best scoring N rules. If no improvement is made to the collection of rules, the search is stopped. In principle, our procedure supports any rule scoring function. Numerous popular rule scoring functions (for discrete targets) are available: χ^2, precision, *WRAcc* [67], leverage and lift. After the induction phase, the significance of the findings is tested using the Fisher's exact test [69]. To cope with the multiple-hypothesis testing problem, Hedwig supports Holm-Bonferroni [70] direct adjustment method to control the familywise error rate (FWER) and the Benjamini-Hochberg-Yekutieli [71] to control the false discovery rate (FDR).

4.4.2 Experimental evaluation

This section compares the Hedwig algorithm performance with the algorithms presented in Section 4.3. We followed the same experimental setting for Hedwig and compared the results.

In our previous work [17] we compared SEGS, SDM-SEGS and SDM-Aleph approaches on two datasets: ALL and hMSC and used the Friedman test together with the Nemenyi post-hoc test to determine which algorithm performs significantly better compared to the others. The methodology used follows the suggestions by Demšar [72].

In this work, we employ the comparison methodology by [73], which is an extension of the paper by Demšar [72]. In particular, we use the Iman-Davenport test, together with Hommel's post-hoc test to determine the "statistical ranks" of the four algorithms for each subgroup discovery evaluation measure.

Table 4.2 presents the statistical rank for each algorithm on each dataset and each evaluation measure. If two algorithms have the same rank (e.g., both achieve rank 1),

[5] The new variable needs to be 'consumed' by a literal to be conjunctively added to this clause in the next step of rule refinement.

there is no statistically significant difference between them for that combination of dataset and measure.

For each algorithm, we counted the number of wins. Our latest approach Hedwig acquired the most wins (5 out of 10), although the other approaches were not far behind (SDM-Aleph with 4 out of 10, SDM-SEGS and SEGS with 3 out of 10).

4.5 Software Availability

In this section we describe the Semantic Data Mining software developed as part of this thesis.

4.5.1 SDM-SEGS and SDM-Aleph

SDM-SEGS and SDM-Aleph [17] were the first two SDM systems developed in this thesis, both described in Chapter 4.

Code

Both systems are open-source and available to be used as web-services. We provide widgets for Orange4WS [74] and ClowdFlows. The complete code repository (including web-service wrappers and Orange widgets) is available on GitHub[6].

Workflows

We provide example workflows for both systems on the public installation of ClowdFlows[7]:

- an SDM-Aleph example workflow shown in Figure 4.4,
- an SDM-SEGS example workflow shown in Figure 4.5.

4.5.2 Hedwig

Hedwig is the successor of SDM-Aleph and SDM-SEGS. It attempts to take the best from both systems.

Code

The Hedwig tool is implemented as a Python command-line tool and library. It is also available as a ClowdFlows widget. The tool is open source and is available on GitHub[8], together with usage examples.

Workflows

We provide an example for using Hedwig on the public installation of ClowdFlows[9]; the example workflow is shown in Figure 4.6.

[6] https://github.com/anzev/sdmtoolkit
[7] http://clowdflows.com/existing-workflows/
[8] https://github.com/anzev/hedwig
[9] http://clowdflows.com/existing-workflows/

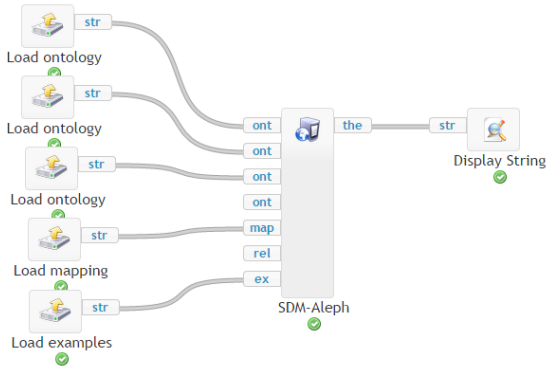


Figure 4.4: An SDM-Aleph example workflow, available at `http://clowdflows.com/workflows/680/`.

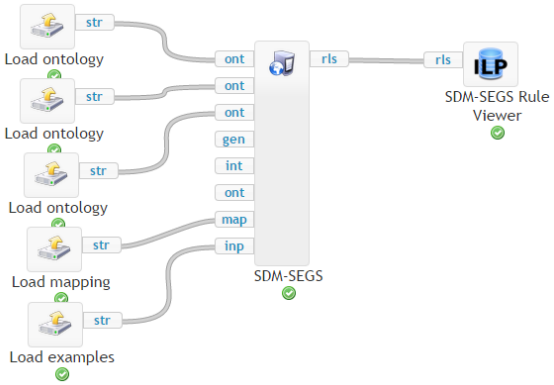


Figure 4.5: An SDM-SEGS example workflow, available at `http://clowdflows.com/workflows/575/`.

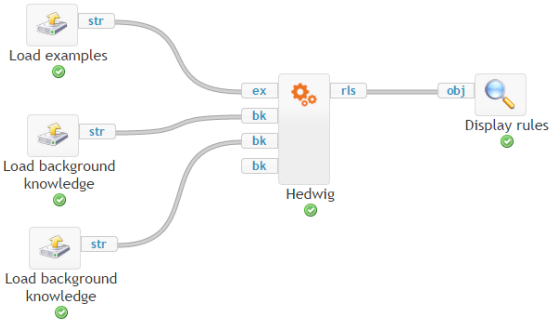


Figure 4.6: A Hedwig example workflow, available at `http://clowdflows.com/workflow/7031/`.

4.6 Related Publication

Our first journal publication on semantic data mining is the following publication (included in this chapter):

A. Vavpetič and N. Lavrač, "Semantic subgroup discovery systems and workflows in the SDM-toolkit," *The Computer Journal*, vol. 56, no. 3, pp. 304–320, 2013.

The author's contributions are as follows. Anže Vavpetič designed, ran the experiments, and implemented the software. Nada Lavrač contributed the idea of semantic data mining and led the project of implementing the scientific workflows in Orange4WS. Both authors contributed to the text of the publication.

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.

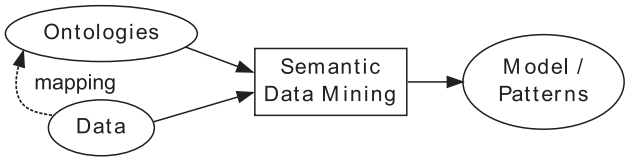


FIGURE 1. Schema of the semantic data mining task, with ontologies and annotated data as inputs.

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

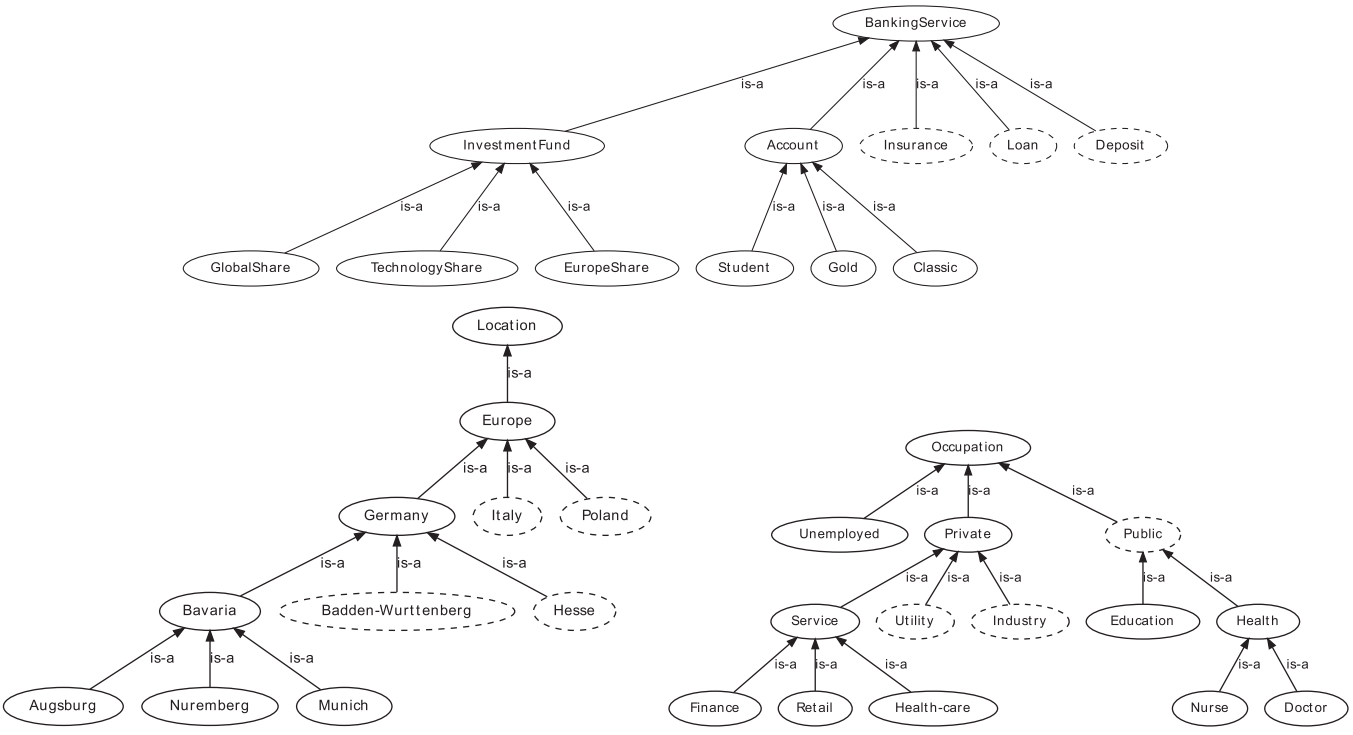


FIGURE 4. The ontologies of banking services, locations and occupations. Concepts with omitted sub-concepts are drawn with a dashed line.

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.

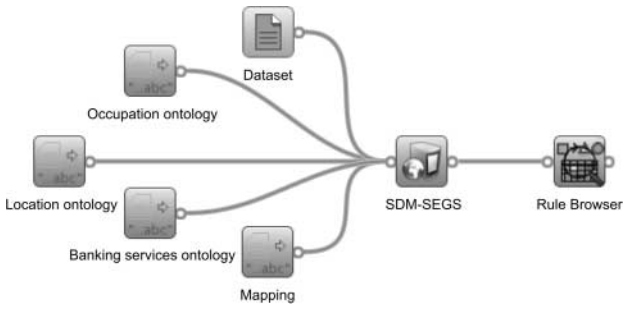
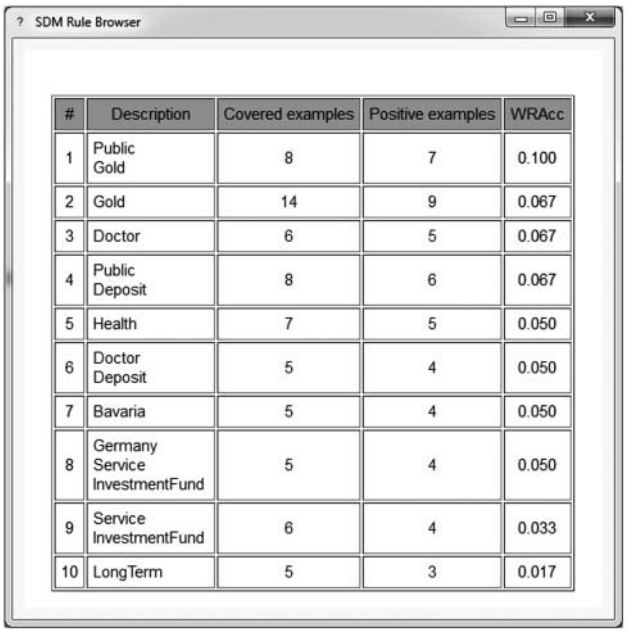


FIGURE 6. A workflow in the SDM-Toolkit for solving the banking problem.

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.

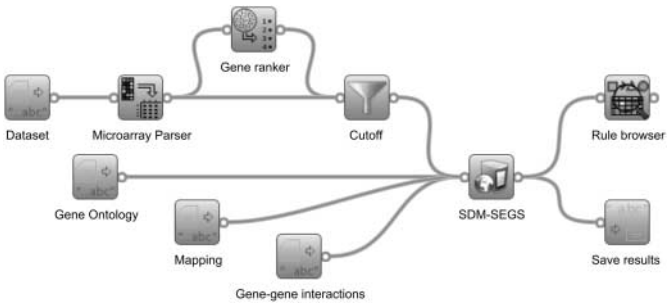


FIGURE 8. A workflow in the SDM-Toolkit for knowledge discovery from microarray data.

```
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

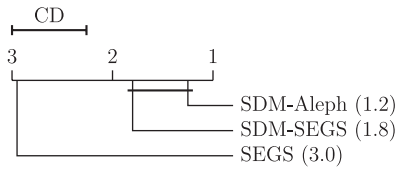


FIGURE 10. Example CD diagram for comparing the algorithms on the hMSC domain for the average support measure, $\alpha = 0.05$.

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

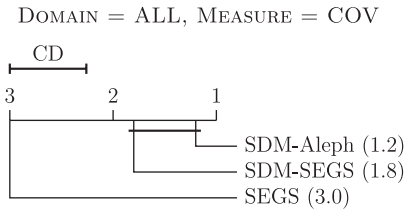


FIGURE 11. CDs between the algorithms on the ALL domain for measure COV, $\alpha = 0.05$.

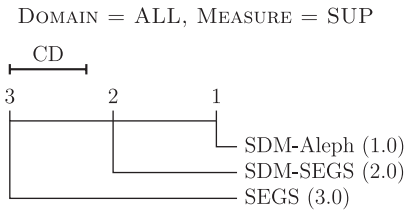


FIGURE 12. CDs between the algorithms on the ALL domain for measure SUP, $\alpha = 0.05$.

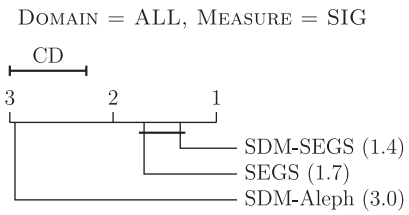


FIGURE 13. CDs between the algorithms on the ALL domain for measure SIG, $\alpha = 0.05$.

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

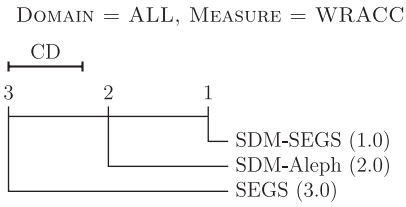


FIGURE 14. CDs between the algorithms on the ALL domain for measure WRACC, $\alpha = 0.05$.

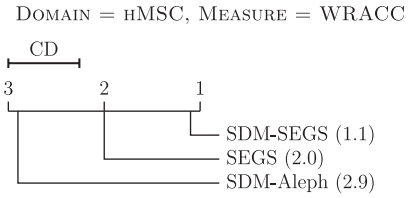


FIGURE 15. CDs between the algorithms on the hMSC domain for measure WRACC, $\alpha = 0.05$.

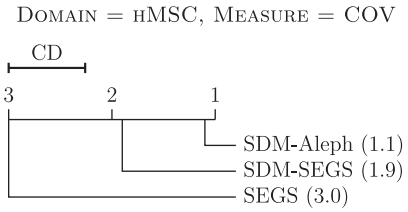


FIGURE 16. CDs between the algorithms on the hMSC domain for measure COV, $\alpha = 0.05$.

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

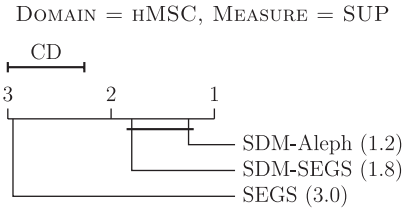


FIGURE 17. CDs between the algorithms on the hMSC domain for measure SUP, $\alpha = 0.05$.

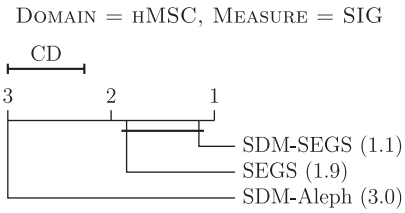


FIGURE 18. CDs between the algorithms on the hMSC domain for measure SIG, $\alpha = 0.05$.

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

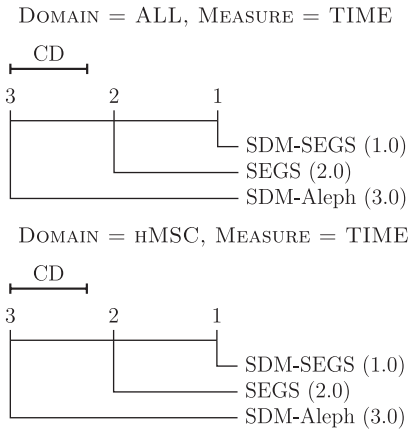


FIGURE 19. CD diagrams for runtime on both domains, $\alpha = 0.05$.

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.

Chapter 5

Semantic Subgroup Discovery Applications

In this chapter we present several applications of semantic subgroup discovery: explaining subgroups of breast cancer patients, characterizing cancer types using multi-resolution DNA aberration data, and analysis of financial news articles.

5.1 Explaining Subgroups of Breast Cancer Patients

This section presents an application of semantic subgroup discovery that helps to explain groups of instances through domain vocabulary encoded in ontologies. We briefly describe the methodology, experiment, and results; further details are available in the included journal publication.

5.1.1 Methodology

Semantic subgroup discovery approaches such as SEGS, SDM-SEGS, SDM-Aleph and Hedwig can serve as explanatory subsystems in the presented methodology to semantically describe and explain contrasting groups in input data.

These are the steps of the methodology:

Step 1 The first step involves finding relevant sets of instances (relevant to the user) by, for example, applying a subgroup discovery algorithm, thus creating a new labeling for the instances in terms of their subgroup membership.

Step 2 The second step deals with ranking the attributes (e.g., using ReliefF) according to their ability to distinguish between the subgroups.

Step 3 The third step of the methodology induces symbolic explanations of a selected target set of instances (subgroup detected in the first step) by using ontological concepts. This step involves using a SDM approach.

We must emphasize that the methodology consists of several steps, which are not novel by themselves, but are used in a novel fashion; also, each step of the components can be easily interchanged with several alternatives. The details of the methodology are available in the journal publication at the end of this chapter.

5.1.2 Experimental results

This section presents and discusses the application of the presented methodology on gene expression data. More specifically, we evaluate the methodology on a breast cancer dataset using our implementation of the methodology as a workflow in the ClowdFlows platform.

The gene expression dataset used in our analysis is the dataset published by Sotiriou et al. [75] of 12,718 genes and 189 patients. The ultimate goal of the experiment was to induce meaningful high-level semantic descriptions of subgroups found in the data which could provide important information in the clinical decision making process.

We first employed subgroup discovery on the 0/1 gene expression data, next we ranked the genes according to a selected subgroup of instances, and lastly, we generated explanations using GO and KEGG vocabulary. We showed that by using our methodology one can essentially automatically reproduce the observations noted in the earlier work by Sotiriou et al. This can encourage the researchers to apply the presented methodology in similar exploratory analytics tasks. The following journal publication lists the details of the experiment, as well as a motivational example and a comparison with a competing method.

5.1.3 Related publication

Details of the methodology and experiments can be found in the following journal article (included at the end of this section):

A. Vavpetič, V. Podpečan, and N. Lavrač, "Semantic subgroup explanations," *J. Intell. Inf. Syst.*, vol. 42, no. 2, pp. 233–254, 2014.

The author's contributions are as follows. Anže Vavpetič designed, ran the experiments, and implemented the software. Vid Podpečan contributed to the scientific workflows and related work, while Nada Lavrač contributed to the idea of using semantic data mining for explanations, as well as to the scientific workflows. All authors contributed equally to the methodology and experimental design. All authors contributed to the text of the publication.

J Intell Inf Syst
DOI 10.1007/s10844-013-0292-1

Semantic subgroup explanations

Anže Vavpetič · Vid Podpečan · Nada Lavrač

Received: 8 May 2013 / Revised: 20 September 2013 / Accepted: 19 November 2013

Abstract Subgroup discovery (SD) methods can be used to find interesting subsets of objects of a given class. While subgroup describing rules are themselves good explanations of the subgroups, domain ontologies can provide additional descriptions to data and alternative explanations of the constructed rules. Such explanations in terms of higher level ontology concepts have the potential of providing new insights into the domain of investigation. We show that this additional explanatory power can be ensured by using recently developed semantic SD methods. We present a new approach to explaining subgroups through ontologies and demonstrate its utility on a motivational use case and on a gene expression profiling use case where groups of patients, identified through SD in terms of gene expression, are further explained through concepts from the Gene Ontology and KEGG orthology. We qualitatively compare the methodology with the supporting factors technique for characterizing subgroups. The developed tools are implemented within a new browser-based data mining platform ClowdFlows.

Keywords Data mining · Semantic data mining · Subgroup discovery · Ontologies · Microarray data

A. Vavpetič (✉) · V. Podpečan · N. Lavrač
Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia
e-mail: anze.vavpetic@ijs.si

V. Podpečan
e-mail: vid.podpecan@ijs.si

N. Lavrač
e-mail: nada.lavrac@ijs.si

A. Vavpetič · N. Lavrač
Jožef Stefan International Postgraduate School, Ljubljana, Slovenia

N. Lavrač
University of Nova Gorica, Nova Gorica, Slovenia

Published online: 06 December 2013

1 Introduction

The paper first addresses the task of subgroup discovery, initially defined by Klösgen (1996) and Wrobel (1997), which is based both on classification and association discovery approaches. The goal is to find subgroups of individuals that are statistically important according to some property of interest of a given population of individuals. For example, a subgroup should be as large as possible and exhibit the most unusual distribution of the target class compared to the rest of the population.

Subgroup discovery methods can be used to find descriptions of objects of a given class—in binary as well as in multi-class problems. Subgroup descriptions, formed as rules with a class label in the rule conclusion and a conjunction of attribute values in the rule condition, typically provide sufficiently informative explanations of the discovered subgroups. However, with the expansion of the Semantic Web and the availability of numerous domain ontologies which provide domain background knowledge and semantic descriptors to the data, we are faced with the challenge of using this publicly available information also to provide explanations of rules initially discovered by standard symbolic data mining and machine learning algorithms. Approaches which would enhance symbolic rule learning with the capability of providing explanations of the rules also in terms of higher-level concepts than those used in rule descriptors, have a potential of providing new insights into the domain of investigation.

To give a simple example, suppose a standard subgroup discovery algorithm produces two rules for a dataset with patients (with the class *cancer*=0/1) and genes as attributes:

$$R_1 : (cancer = 1) \leftarrow (g_A = 1) \wedge (g_B = 1) \wedge (g_C = 0)$$
$$R_2 : (cancer = 1) \leftarrow (g_A = 0) \wedge (g_B = 1) \wedge (g_D = 1)$$

Each rule defines a subgroup of patients for which the right-hand side is true. These rules are by themselves explanatory in terms of single genes. But due to the existence of genetic regulatory networks, there are complex dependency structures between genes, e.g., multiple genes might be associated with a certain biological function. Using an ontology of biological knowledge (see next paragraph), we can find higher-level patterns on top of the gene-level patterns (such as rules R_1 and R_2). We propose that this can be achieved, for example, by taking R_1 and R_2 as the new classes and inducing new higher-level patterns by grouping the single genes into higher-level concepts defined by the ontology. An example higher-level rule E_1 (which we call an *explanation*) is:

$$E_1 : (cls = R_1) \leftarrow (c_1 = 1)$$

E_1 states that the patients defined by subgroup R_1 (the new target class) are characterized by the higher-level concept c_1 (e.g., a biological function), in contrast to patients from R_2. This is a higher-level statement, which takes into account multiple genes which are associated with the particular biological function c_1. This association knowledge is provided beforehand by the domain ontology.

We must emphasize that this explanatory step is not limited only to subgroup discovery. Essentially, the explanatory stage can be applied on any sets of examples that are of interest to the user, provided that a suitable ontology exists.

In this paper we show that such an additional explanatory step can be performed by using recently developed semantic subgroup discovery approaches (Podpečan et al. 2011a; Vavpetič and Lavrač 2013). The new methodology is show-cased on two use cases: a motivational use case of bank customers and on a gene expression profiling real-life use case.

The motivational use case showcases the methodology on a simple use case with banking customers and three simple ontologies, in order to illustrate the steps of the methodology.

In the gene expression use case, groups of patients of a selected grade of breast cancer, identified through subgroup discovery in terms of gene expression, are further explained through terms from the Gene Ontology[1] (GO) and Kyoto Encyclopedia of Genes and Genomes[2] (KEGG) and Entrez[3] gene-gene interaction data. The motivation for the use case in breast cancer patient analysis comes from the experts' assumption that there are several subtypes of breast cancer. Hence, in addition to distinguish between patients with breast cancer (the positive cases) and healthy patients, the challenge is first to identify breast cancer subtypes by finding subgroups of patients followed by inducing explanations in terms of identical biological functions, processes and pathways of genes, characterizing different molecular subtypes of breast cancer.

With the two use cases we demonstrate that the proposed approach is general and can be applied in any application area, provided the existence of domain specific ontologies.

The main contributions of the present work are as follows. First, inducing explanations of subgroups (or, e.g., clusters of instance), regardless of how the subgroups were detected, in terms of knowledge encoded in a domain ontology. Second, we have made our approach readily available on the web, as a reusable data mining workflow, which we hope will be a valuable resource for scientists, enabling them to use the workflow on new data, as well as adapt it for other use cases.

In addition, this work upgrades our early results (Vavpetič et al. 2012) in several ways. First, we have fully integrated our approach with the microarray analysis SegMine system (Podpečan et al. 2011a). Researchers using our tools can now also use the results of our methodology to query the Biomine search engine (Eronen and Toivonen 2012). Biomine essentially merges a large number of public biological databases into a common graph. The nodes in this graph are biological entities, while the edges are relations between them. Biomine offers advanced probabilistic graph search algorithms that can discover the parts of the graph most relevant to the given query. Examples of queries are: finding a neighborhood of a set of nodes or a graph connecting two sets of nodes. Biomine also offers a visualization tool for the user to explore the resulting subgraph.

Next, compared to our previous work where we made our tools available in the Orange4WS (Podpečan et al. 2011b) data mining platform, we have now moved to a new browser-based platform ClowdFlows (Kranjc et al. 2012). The main benefits of moving to ClowdFlows are: (a) no installation is required prior to using our tools (apart from an internet connection and a web browser), (b) scientific workflows and data can be shared by sharing a single URL, and (c) users can easily clone and adapt existing workflows to their own needs. We give an overview of the implementation, as well as discuss the pros and cons of the approach. In addition, the related work and the methodology are described in much more detail, enabling detailed methodology understanding and enabling its modification (upgrades by other researchers).

Additionally, the paper shows that the methodology is generally applicable for explaining groups of instances in any domain in which domain concepts are organized into ontologies and where data descriptions (attributes or attribute values) correspond to concepts from the ontologies. This is demonstrated with the two distinct use cases.

[1] http://www.geneontology.org/

[2] http://www.genome.jp/kegg/

[3] http://www.ncbi.nlm.nih.gov/sites/gquery

Finally, we compare our approach to the related *supporting factors* (Gamberger and Lavrač 2003) methodology, which is also used to characterize subgroups and can be of great help to the interpretation of subgroup discovery patterns of domain experts. The experiments show that supporting factors are more useful when concentrating on specific low-level attributes or features is desirable, but when more general descriptions are needed, they are not as easy to interpret as the method presented in this paper. This restriction is more apparent on gene expression data, since the supporting factors are given in terms of genes.

The paper is structured as follows. Section 2 discusses the related work. The proposed methodology is presented in detail in Section 3. The ClowdFlows platform and the implementation of the methodology are described in Section 4. In Section 5 the methodology is applied to two use case scenarios: a motivational use case and a use case with breast cancer gene expression (microarray). The presented methodology is also compared to the supporting factors methodology on the second use case. Section 6 concludes the paper and presents the plans for further work.

2 Related work

This section discusses the work related to the main steps of the proposed methodology. Given a complex multi-step approach, the related work covers subgroup discovery, contrast mining, subgroup explanation, and semantic data mining. Mining of enriched gene sets from gene expression data is also relevant for the biomedical use case presented in Section 5 (analysis of breast cancer data) which is used to evaluate the proposed methodology.

Subgroup discovery The problem of subgroup discovery was defined by Klösgen (1996) and Wrobel (1997) as search for population subgroups which are statistically interesting and which exhibit unusual distributional characteristics with respect to the property of interest. Subgroup descriptions are conjunctions of attributes and values which characterize the selected class of individuals. Several algorithms were developed for mining interesting subgroups using exhaustive search or using heuristic approaches: Explora (Klösgen 1996), APRIORI-C (Jovanoski and Lavrač 2001), APRIORI-SD (Kavšek and Lavrač 2006), SD-Map (Atzmüller and Puppe 2006), SD (Gamberger and Lavrač 2002), CN2-SD (Lavrač et al. 2004). These algorithms employ different heuristics to asses the interestingness of the discovered rules, which is usually defined in terms of rule unusualness and size.

Contrast mining Mining of contrasts in data has been recognized as one of the the fundamental tasks in data mining (Webb et al. 2003). The underlying idea is to discover and understand contrasts (differences) between objects of different classes, different time periods, spatial locations, objects within a class or various combinations of these. One of the first algorithms which has explicitly addressed the task of mining contrast sets is the STUCCO algorithm, developed by Bay and Pazzani (2001). It searches for conjunctions of attributes and values (contrast sets) which exhibit different levels of support in mutually exclusive groups, STUCCO enforces statistically sound results by employing testing for statistical significance and p-value correction along with minimum support threshold. Mining for contrasting sets is also related to exception rule mining as defined by Suzuki (1997, 2006) where the goal is to discover rare deviating patterns which complement strong base rules to form rule pairs. Suzuki (2006) defines an exception as something different from most of the rest of the data which can be also seen as a contrast to given data and/or existing

domain knowledge. A recent approach developed by Langohr et al. (2013), which proposes *contrasting subgroup discovery*, is closely related to the methodology presented in this paper. It extends classical subgroup discovery using a three-step approach and allows for the discovery of subgroups which cannot be found with classical subgroup discovery. Two subgroup discovery steps are complemented by the intermediate, contrast set definition step. In this intermediate step, the user explicitly defines two contrasting classes using set-theoretic functions and the subgroups discovered in the first step. In this way, generalized subgroups consisting of members from different classes can be discovered. While the approach is general and can be used on any data, it is especially well-suited for domains such as systems biology and biomedicine where comparing e.g., different time points in experimental data or several subtypes of a disease is a typical task.

Subgroup explanation The need of developing methods for presenting contrast sets to the user has already been recognized by Gamberger and Lavrač (2002) and Webb et al. (2003). Kralj Novak et al. (2009) have shown that contrast set mining, emerging pattern mining (Dong and Li 1999) as well as subgroup discovery can be viewed as variants of rule learning by providing appropriate definitions of compatibility; they also presented several subgroup visualization approaches, enabling subgroup comparison in terms of their size and distributional unusualness. However, to the best of our knowledge, neither different subgroup discovery algorithms nor the relatively efficient contrast/exceptional pattern mining algorithms like STUCCO (Bay and Pazzani 2001) and PEDRE (Suzuki 1997) address the representation and explanation of subgroups/contrasts using the available background knowledge and ontologies.

Semantic data mining While subgroup descriptions in the form of rules are relatively good descriptions of subgroups there is also abundance of background knowledge in the form of taxonomies and ontologies readily available to be incorporated to provide better high-level descriptions and explanations of discovered subgroups. Especially in the domain of systems biology the GO ontology, KEGG orthology and Entrez gene-gene interaction data are good examples of structured domain knowledge. The challenge of incorporating domain ontologies in data mining was addressed in the recent work on semantic data mining (SDM) (Hilario et al. 2011; Lavrač et al. 2011; Lawrynowicz and Potoniec 2011; Vavpetič and Lavrač 2013; Žáková et al. 2006).

Using a data mining ontology for meta-learning has been proposed in Hilario et al. (2011). 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.

In Lawrynowicz and Potoniec (2011), 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 $\mathcal{EL}^{++}$ description logic.

In Žáková et al. (2006) an engineering ontology of CAD (Computer-Aided Design) elements and structures is used as background knowledge to extract frequent product design patterns in CAD repositories and discovering predictive rules from CAD data.

This work is built upon the SDM toolkit developed by Vavpetič and Lavrač (2013). The toolkit includes two semantic data mining systems: SDM-SEGS and SDM-Aleph. SDM-SEGS is an extension of the earlier domain-specific algorithm SEGS (Trajkovski et al.

2008) which allows for semantic subgroup discovery in gene expression data. SEGS constructs gene sets as combinations of GO ontology terms, KEGG orthology terms, and terms describing gene-gene interactions obtained from the Entrez (Maglott et al. 2005) database. SDM-SEGS extends and generalizes this approach by allowing the user to input any set of ontologies in the OWL format and an empirical data collection which is annotated by domain ontology terms. SDM-SEGS employs ontologies to constrain and guide the top-down search of a hierarchically structured space of induced hypotheses. SDM-Aleph, which is built using the popular ILP system Aleph (Srinivasan 2007) does not have the limitations of SDM-SEGS, imposed by the domain-specific algorithm SEGS, and can accept any number of OWL ontologies as background knowledge which is then used in the learning process.

Semantic data mining and link discovery in enriched gene set analysis In the domain of systems biology, the SegMine methodology (Podpečan et al. 2011a) enables semantic analysis of microarray data by integrating the SEGS algorithm, GO and KEGG, and the Biomine system which integrates several public databases with a sophisticated algorithm for link discovery. Parts of the SegMine methodology can be reused in the methodology proposed in this paper for the specific use case of gene expression profiling. For example, link discovery can provide additional and potentially new information about the discovered important genes, subgroups and ontology terms.

Characterizing outliers In Angiulli et al. (2013), they consider a related task of characterizing attributes that account for a small group of anomalous examples-outliers. They define the notion of exceptional property and exceptionality score. They are designed to work especially with small samples. In contrast to our work, they focus mainly on small, anomalous groups of examples. The second main difference is that they do not try to generalize over the given attributes, since the exceptional properties are in terms of the original attributes.

Supporting factors The most relevant related work is the work by Gamberger and Lavrač (2003). In their work, they deal with characterizing subgroups through *supporting factors*. Supporting factors are features with significantly different value distributions that are not part of the subgroup description. Supporting factors are important, e.g., for medical decision making, which requires as much supportive evidence as possible. We compare our methodology with supporting factors in Section 5.2.

3 Methodology

Semantic subgroup discovery approaches such as SEGS, SDM-SEGS and SDM-Aleph can serve as explanatory subsystems in the presented methodology to semantically describe and explain contrasting groups in input data. This section presents the steps of the proposed methodology. The first step involves finding relevant sets of instances (relevant to the user) by applying a subgroup discovery algorithm, thus creating a new labeling for the instances in terms of their subgroup membership. The second step deals with ranking the attributes according to their ability to distinguish between the subgroups. The third step of the methodology induces symbolic explanations of a selected target set of instances (subgroup detected in the first step) by using ontological concepts.

We must emphasize again that the methodology consists of several steps, which are not novel by themselves, but are used in a novel fashion; also, each step of the components can be easily interchanged with several alternatives.

3.1 Identifying interesting sets of instances and creating a new labeling

To find a potentially interesting set of instances, the user can choose from a number of data mining algorithms. Data mining platforms such as Weka (Hall et al. 2009), Orange (Demšar et al. 2004), Orange4WS (Podpečan et al. 2009) and ClowdFlows (Kranjc et al. 2012) offer various clustering, classification and visualization techniques. A potentially interesting set of instances can be a cluster of instances, instances in a node of a decision tree, a set of instances revealed by a visualization method, a set of instances covered by a subgroup description, and others; in the following paragraphs we concentrate on subgroup discovery, but other techniques that define some sort of sets of examples can be used analogously (e.g., clustering; the user chooses between clusters instead of subgroups).

First, some basic notation needs to be established. Let $D = \{e_1, e_2, \ldots, e_n\}$ be a dataset of classified instances, called examples in the rest of this paper. Examples are defined by values of a set of attributes $A = \{a_1, a_2, \ldots, a_m\}$ and a continuous or discrete target variable y (note that unsupervised methods do not require a target variable). Let v_{ij} denote the value of attribute a_j for example e_i.

In the following, subgroups and clusters are represented as sets of examples. Let S_A and S_B denote two sets of examples ($S_A \cup S_B \subseteq D$) that are of interest to the user who wants to determine which groups of attributes (expressed as ontological concepts) differentiate S_A from S_B. Note that for subgroup descriptions the following must also hold: $S_A \cap S_B = \emptyset$, since it is typical that subgroups can overlap. This condition is of course not necessary for other settings like clustering.

Regardless of how S_A and S_B have been constructed, the new re-labeled dataset D' is formed as follows. The target variable y is replaced by a binary target variable y' and for each example e_i the new label c' is defined as:

$$c' = \begin{cases} 1, & \text{if } e_i \in S_A \\ 0, & \text{if } e_i \in S_B \end{cases}$$

Note that if D is unlabeled, the new target variable y' is added to the domain. We now illustrate how to determine S_A and S_B using a subgroup discovery (SD) approach.

SD algorithms induce symbolic subgroup descriptions of the form

$$(y = c) \leftarrow t_1 \wedge t_2 \wedge \ldots t_l$$

where t_j is a conjunct of the form $(a_i = v_{ij})$. If a_i is continuous and the selected subgroup discovery algorithm can deal with continuous attributes, t_i can also be defined as an interval such that $(a_i \geq v_{ij})$ or $(a_i \leq v_{ij})$. An example subgroup description constructed from the well known UCI lenses [4] dataset is:

$$(lenses = hard) \leftarrow (prescription = myope) \wedge$$
$$(astigmatic = yes) \wedge (tear\_rate = normal)$$

A subgroup description R can be also viewed as a set of constraints (conjuncts t_i) on the dataset, and the corresponding subgroup as a set of examples *cov(R)* which satisfy the constraints, i.e., examples covered by rule R.

[4] http://archive.ics.uci.edu/ml/datasets/Lenses

If the user is presented with a set of subgroup descriptions $R = \{R_1, R_2, \ldots, R_k\}$, then the set of examples S_A can be defined as $S_A = cov(R_i)$. For subgroup discovery S_B typically represents all other examples $S_B = D \setminus S_A$, because subgroups often overlap. For clustering S_B can be a single other cluster or a union of several clusters, depending on the user's preference.

To give a trivial example, suppose the subgroup discovery procedure returns three subgroup descriptions $R = \{R_1, R_2, R_3\}$ on the previously mentioned UCI lenses dataset. These are as follows:

$$R_1: \ (lenses = hard) \leftarrow (age = young)$$
$$R_2: \ (lenses = soft) \leftarrow (astigmatic = no)$$
$$R_3: \ (lenses = none) \leftarrow (prescription = hypermetrope)$$

For example, R_1 covers all examples that have the attribute-value $age = young$; these examples constitute the rule's *coverage*.

The user can then select S_A and S_B, to give an example, as follows $S_A = cov(R_1)$ and $S_B = D \setminus S_A$. In this scenario S_A contains examples covered by R_1 and S_B contains all examples not covered by R_1.

3.2 Ranking of attributes

Once the re-labeled dataset D' is available, the attributes are assigned ranks according to their ability to distinguish between the two sets of examples S_A and S_B. The resulting ordered attributes and their scores will be used as input examples in the next step of the methodology. The generalizations of the attributes made via the ontological background knowledge will be the constituents of the resulting explanations.

To calculate the ranks, any attribute quality measure can be used, but in practice attribute ranking using the ReliefF (Robnik-Šikonja and Kononenko 2003) algorithm has proven to yield reliable scores for this methodology to work. In contrast to myopic measures (e.g., Gain Ratio), ReliefF takes into account the context of other attributes when evaluating an attribute. This is an important benefit when applying this methodology to datasets such as microarray data since it is known that there are dependencies among many genes.

The ReliefF algorithm works as follows. A random subset of examples of size $m \leq n$ is chosen. Each attribute starts with a ReliefF score of 0. For each randomly selected example e_i and each class c, k nearest examples are selected. The algorithm then goes through each attribute a_l and nearest neighbor e_j $(i \neq j)$, and updates the score of the attribute as follows:

- if e_i and e_j belong to the same class and at the same time have different values of a_l, then the attribute's score is decreased;
- if the examples have different attribute values and belong to different classes, then the attribute's score is increased.

This step of the methodology results in a list of ReliefF attribute scores $L = [(a_1, r_1), \ldots, (a_m, r_m)]$ where r_i is the ReliefF score representing the ability of attribute a_i to distinguish between sets S_A and S_B.

3.3 Inducing explanations using ontologies

At this stage of the methodology a semantic subgroup discovery algorithm (Lavrač et al. 2011; Vavpetič and Lavrač 2013) is applied to generate explanations using the list of ranked attributes L.

First, we need to formalize the notion of an ontology. An ontology is a conceptualization of a certain domain in terms of *concepts* and *relationships* between these concepts. An ontology is a directed acyclic graph, i.e., with no paths starting and ending on the same vertex, with concepts $C = \{c_1, c_2, \ldots c_n\}$ as vertices and relations $R = \{r_1, r_2, \ldots r_m\}$ as edges. Each relation is defined as a set of pairs of concepts: $r_i = \{(c_j, c_k)|c_j, c_k \in C\}$. Commonly used relationships are *subClassOf* (commonly referred to as *is-a*) and *partOf*. In this section we use the Gene Ontology as an example, which uses only these two relations.

Concepts and relations consistute the so-called *T-box* (the terminology). In order to connect the data (ranked attributes) to the ontology, we also require the *A-box* (the assertions). These we can view as a mapping $M = \{(a_i, c_j)|a_i \in A, c_j \in C\}$ of objects (in our case, attributes) onto concepts from the ontology. In the case of the Gene Ontology use case, the gene annotations represent our A-box, which defines which genes are annotated by which ontological concept (e.g., a biological function).

Each subgroup description (rule) induced by a semantic subgroup discovery algorithm represents one explanation, and each explanation is a conjunction of ontological concepts. The assumption here is that a domain ontology $O =< C, R >$ is available and that a mapping M between the attributes (or attribute values; we assume attributes in the rest of this section) and ontological concepts exists. For example, in the case of microarray data, an attribute (gene) IDH1 is mapped to (annotated by) the ontological concept *Isocitrate metabolic process* from the Gene Ontology, indicating that this gene takes part in this particular biological process. Thus, when translated into our methodology, each ontological concept, as well as each explanation, defines a set of attributes.

In other words, an existing semantic subgroup discovery algorithm is at this stage applied in a novel way - the algorithm internals are identical compared to when used for a standard subgroup discovery task.

Annotations enable the explanations to have strictly defined semantics, and from a data mining perspective, this information enables the algorithm to generalize better than by using attribute values alone. The explanations can be made even richer if additional relations among the attributes (or ontological concepts) are included in the explanations. Using the microarray example, genes are known to *interact*, and this information can be directly used to form explanations.

Currently, there are four publicly available SDM systems that can be used for the purpose of inducing explanations:

- SEGS (Trajkovski et al. 2008), a domain specific system for analyzing microarray data using the Gene Ontology, KEGG orthology, and Entrez gene-gene interactions,
- SDM-SEGS (Vavpetič and Lavrač 2013), the general purpose version of SEGS, that enables the use of OWL ontologies, but is limited to a maximum of four ontologies, i.e., the user needs to specify the rule language by defining up to four new roots of their ontology,
- SDM-Aleph (Vavpetič and Lavrač 2013), a general purpose SDM system based on the ILP system Aleph, that can use any number of OWL ontologies,
- Hedwig (Vavpetič et al. 2013), a new subgroup discovery SDM system, which builds upon the benefits of both SDM-SEGS and SDM-Aleph. Namely, it support the full RDFS ontology language and exploits the *subClassof* hierarchy to efficiently structure the search space.

All four systems focus on inducing explanations in the form of rules with conjuncts corresponding to ontological concepts. To illustrate how explanations are induced, consider that SEGS or SDM-SEGS (they have the same rule construction algorithm, but different

rule selection process) is selected to be used on a microarray domain. The algorithm used by SEGS and SDM-SEGS is the simplest of the four and is good for illustrating the semantic nature of the learning process, but it has its drawbacks. Namely, due to its simplicity only the *subClassOf* relation is exploited and one additional relation between the genes/attributes. SDM-Aleph is similar, except that it imposes no restrictions on the number of relations. On the other hand, Hedwig has no such limitations. The background knowledge can contain arbitrary relations, with *subClassOf* having a special status in that it is exploited to structure the search space.

Note that in the following description, genes can be thought of as instances or examples, since the algorithm is not limited only to genes. The idea behind SEGS as well as SDM-SEGS, illustrated on the problem of finding explanations for top-ranked genes, is as follows (Fig. 1 shows the rule construction algorithm in pseudo code).

The set of explanations/subgroup descriptions is constructed using top-down bounded exhaustive search according to the user-defined constraints (e.g., minimum support). The algorithm considers all explanations that can be formed by taking one concept from each ontology as a conjunct.

The input list L of ranked genes is first split into two classes. The set of genes above a selected threshold value is the set of differentially expressed genes for which a set of rules is constructed (these rules describe sets of genes which distinguish set S_A from set S_B).

The construction procedure starts with a default rule $top(X) \leftarrow$, with an empty set of conjuncts in the rule condition, which covers all the genes. With $top(X)$ we denote the target concept, which is in this case a set of attributes that near or at the *top* of the list L—thus good at distinguishing between the two sets. Next, the algorithm tries to conjunctively add the root concept of the first ontology (yielding e.g., $top(X) \leftarrow biological\_process(X)$) and if the new rule satisfies all of the size constraints (`MIN_SIZE` - minimum number

```
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 0 < rule.terms.size < MAX_TERMS:
     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 rule.terms.size < MAX_TERMS:
     rules.add(rule)

 return rules
```

Fig. 1 Rule construction procedure of (SDM-)SEGS

of genes coverd by a rule, `MAX_TERMS` - maximum number of conjunctions in a single rule), it adds it to the rule set and recursively tries to add the root concept of the next ontology (e.g., $top(X) \leftarrow biological\_process(X) \wedge molecular\_function(X)$). In the next step all the child concepts of the current conjunct/concept are considered by recursively calling the procedure. Due to the transitivity 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 rules which would be generated if this rule were further specialized.

Additionally, the user can specify gene interaction data by specifying the *interacts* relation. In this case, 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 $top(X) \leftarrow c_1(X)$ and the algorithm tries to add the term/concept $c_2(X)$, then it also separately tries to append a compound term $interacts(X, Y) \wedge c_2(Y)$.

In SEGS, the constructed explanations are assigned scores using several established methods (e.g., GSEA Subramanian et al. 2005) and the significance of the explanations is evaluated using permutation testing (Trajkovski et al. 2008).

In our setting, the resulting descriptions correspond to subgroups of attributes (e.g., genes) which enable distinguishing between sets S_A and S_B. The interpretation is simple, due to the ontological concepts (conjuncts). Consider the following subgroup description:

$$top(X) \leftarrow immune\_system\_process(X) \wedge plasma\_membrane(X) \wedge interacts(X, Y) \wedge T\_cell\_receptor\_signaling\ pathway(Y).$$

This rule can be interpreted as follows. One of the top groups of genes (attributes) that are capable of distinguishing S_A from S_B, are the genes which take part in the *immune system process*, are part of the *plasma membrane* and interact with genes that are part of the *T cell receptor signaling pathway*.

4 Implementation

The described methodology was implemented in ClowdFlows (Kranjc et al. 2012), a publicly available workflow environment. We have extended the original implementation in the Orange4WS (Podpečan et al. 2011b) platform in order to make the experimental data and workflow, as well as the individual re-usable components easily accessible. As the ClowdFlows user interface runs entirely in a web browser there are no software requirements. Moreover, the developed workflows and the results of their execution can be shared by providing a link to the workflow. In the following we summarize the new implementation along with the most relevant features of ClowdFlows.

4.1 The ClowdFlows platform

ClowdFlows is a new generation platform for data mining which is implemented as a web application. It is based on the concept of *visual programming* which denotes the construction of complex procedures (workflows) from smaller building blocks (widgets) on a *canvas*. ClowdFlows offers a large collection of implemented algorithms, procedures and visualizations from different scientific fields: data mining, natural language processing, text mining, systems biology and inductive logic programming. New components can be implemented in the ClowdFlows server application or can be imported as web services. All included

components are available as widgets and can be used in the construction of data analysis workflows.

Two of the most important features of ClowdFlows are its graphical user interface and the database, which stores all information about components, workflows, data, and results. The graphical user interface, which runs as a web application, allows the user to interactively construct the workflow by placing the appropriate component on the canvas, set their paramenters, connects inputs and outputs and execute them. The database, on the other hand, stores all vital information and enables sharing of the constructed solutions, data, and experimental results by making the workflow accessible under a unique public URL. This greatly simplifies the evaluation of experimental results.

4.2 Implementation of the methodology workflow

The proposed methodology was implemented as a ClowdFlows workflow. Widgets from different ClowdFlows packages (such as utility widgets, e.g., *Load dataset*) as well as several newly developed components were deployed. First, the subgroup discovery package was used (some of these widgets are based on the Subgroup Discovery toolkit for Orange[5]). Second, the SDM-toolkit (Vavpetič and Lavrač 2013) and the SegMine tools (Podpečan et al. 2011a) from our previous work were also moved to ClowdFlows. Having these widgets made available within the platform, we were able to connect them into a workflow implementing our methodology. Figures 3 and 4 show two ClowdFlows workflows using our methodology for two use cases. Since the developed widgets are self-contained units with a well defined task, they can be re-used for other tasks as well (the roles of particular widgets are discussed in more detail Section 5).

5 Use cases

In this section we present the application of our methodology on two use cases. The first is a motivational use case intended to illustrate the methodology as well as showing how it can be applied using the ClowdFlows platform. The second use case is an application on real-world gene expression microarray data. On the second use case, we also apply the related supporting factors approach and qualitatively compare it to our approach.

5.1 Illustrative use case

This subsection further illustrates and motivates the use of the methodology on an easy-to-understand toy use case. First, we describe the dataset and cast the problem in our new framework. Next, we present the workflow developed for solving the toy problem by explaining each of the workflow's components.

This use case is an adaptation of the proof-of-concept semantic data mining dataset from Vavpetič and Lavrač (2013). 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 attributes of the dataset are binary.

[5] http://kt.ijs.si/petra_kralj/SubgroupDiscovery/

Table 1 Table of bank customers described by several attributes and class 'big spender'

Doctor	Nurse	Munich	Rome	Classic	Gold	...	Big spender
1	0	0	0	1	0	...	yes
1	0	0	0	0	1	...	yes
0	0	1	0	0	1	...	yes
1	0	0	0	1	0	...	yes
0	0	0	0	0	1	...	yes
...	...	...	...	...	...	...	
0	0	0	0	0	1	...	no
0	1	0	0	1	0	...	no
0	0	0	0	1	0	...	no
0	0	0	0	0	1	...	no
0	0	0	0	1	0	...	no

For example, the attribute-value pair *Doctor=1* indicates that a particular customer is a doctor. The bank also labeled the clients as 'big spenders' or not and wants to find patterns describing big spenders. Table 1 presents a subset of the training data.

Suppose we also have three ontologies available as background knowledge for this problem: an ontology of banking services, an ontology of locations and an ontology of occupations, shown in Fig. 2. Note that the attributes of the dataset correspond to the leaves of the ontologies.

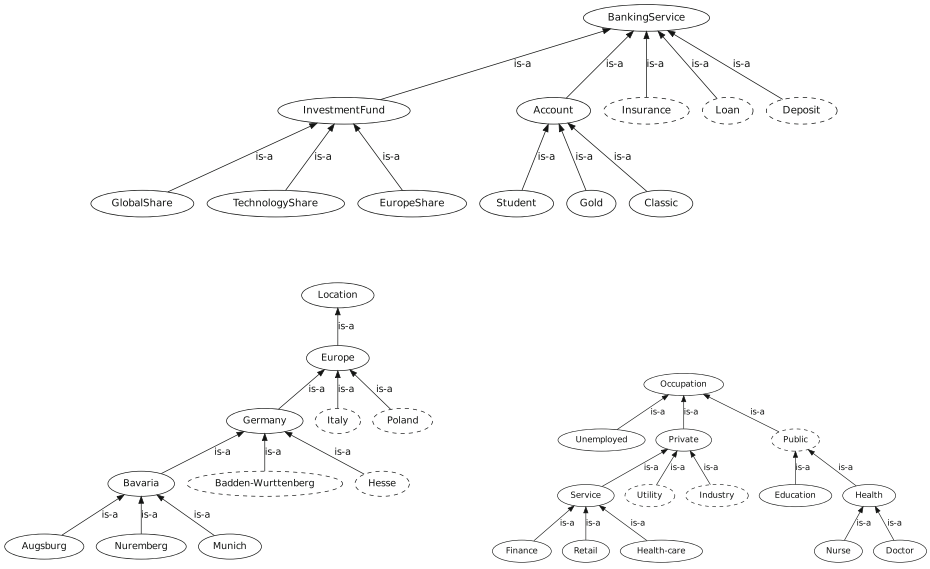


Fig. 2 The ontologies of banking services, locations and occupations. Concepts with omitted sub-concepts are drawn with a *dashed line*

In terms of our methodology, we first want to find descriptions of customers that are big spenders. After finding and selecting an interesting subgroup, we wish to use the knowledge of the domain ontologies to explain what are the differences between this subgroup of customers compared to the other customers.

Figure 3 shows a workflow developed to solve the described problem using our methodology. The workflow neatly follows the steps outlined in Section 3.

Step 1 *Identifying interesting sets of instances and creating a new labeling*, consists of the following components: the dataset is first uploaded (*Load dataset* widget), then standard subgroup discovery is run (*Build subgroups* widget) and the user is prompted to select one or more interesting subgroups (*Select subgroups* widget). The examples are then re-labeled, where the examples in the selected subgroup(s) represent one class, while the rest represents the other class (*Query data with subgroups* and *Table from sets of examples* widgets).

Step 2 *Ranking of attributes*, consists of a single *Ranker* widget, which uses the ReliefF algorithm to assign a score to each of the attributes and outputs a list of pairs (*attribute*, *score*).

Step 3 *Inducing explanations using ontologies*, is composed of one main widget: *SDM-Aleph*. This widget calls the SDM-Aleph web service, which employs the ontologies and the Aleph ILP system to produce subgroups. The widget accepts the list of ranked attributes, the OWL ontologies and the mapping between attribute names and ontology concepts (*Load mapping* and *Load ontology* widgets; note that these are actually *Load file to string* widgets renamed to reflect what they do). The *SDM-Aleph* widget returns a set of subgroups, which is displayed by the *Display subgroups* widget.

This public workflow contains an example experiment (using the dataset described above), where we have used the following settings. In the *Build subgroups* widget we used the SD (Gamberger and Lavrač 2002) subgroup discovery algorithm with 20 % minimum support. In the *Select subgroups* widget we (arbitrarily) selected the subgroup $(Big\ spender = yes) \leftarrow (Cosenza = 0) \wedge (Gold = 1)$. This subgroup contains customers that are not from *Cosenza* and have a *Gold* bank account.

In the *Query data with subgroups, Table from sets of examples* and *Ranker* widgets we used the default settings. In the *SDM-Aleph* widget, we set the data format to 'list' and the cutoff parameter to 10 (this indicates that the input list will be split into two classes by the

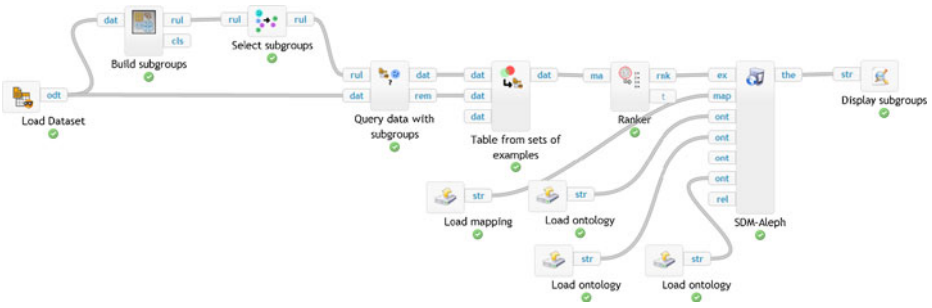


Fig. 3 The workflow implementing the solution to the motivational use case in ClowdFlows. The workflow can be found at http://clowdflows.org/workflow/1283/

Aleph system at the tenth attribute). This is necessary because the *Ranker* widget outputs a *list* of attributes and their scores; alternatively an Orange dataset can be used.

The best scoring subgroup found by *SDM-Aleph* is $top(X) \leftarrow Account(X)$. What is important to note from this simple example is that the *Account* ontological concept does not appear among the dataset attributes (leaves of the ontologies). This means that the explanation was found by generalizing the leaves (*Gold*, *Classic* and *Student* accounts) into the more general concept *Account* (see the first ontology in Fig. 2 to see the relation between these concepts). The explanation indicates that the main difference between the selected subgroup of customers and all other customers is in the type of the account they have. This is intuitive, since the selected subgroup contains the majority of customers with *Gold* accounts, while other customers have either *Classic* or *Student* accounts.

5.2 Biomedical use case

This subsection presents and discusses the application of the presented methodology on gene expression data. More specifically, we evaluate the methodology on the breast cancer dataset using our implementation of the methodology as a workflow in the ClowdFlows platform.

The gene expression dataset used in our analysis is the dataset published by Sotiriou et al. (2006) (GEO series GSE2990). It is a merge of the KJX64 and KJ125 datasets and contains expression values of 12,718 genes from 189 patients with primary operable invasive breast cancer. It also provides 22 metadata attributes such as age, grade, tumor size and survival time. We used the expert-curated re-normalized and binarized version of the dataset from the InSilico database (Taminau et al. 2011). Within the InSilico framework, the raw data was renormalized using fRMA (McCall et al. 2010) and a genetic barcode (0/1) was generated based on whether the expression of a gene was significantly higher (K standard deviations) than the no expression level estimated on a reference of approx. 800 samples. In this setting $g_i = 1$ means that gene g_i is over-expressed and $g_i = 0$ means that it is not. The ultimate goal of the experiment was to induce meaningful high-level semantic descriptions of subgroups found in the data which could provide important information in the clinical decision making process.

Our main motivation for developing the presented methodology is to descriptively characterize various breast cancer subtypes, while in the experiments presented here we focus on describing breast cancer grades, which enables us to focus on the evaluation of the methodology.

The conducted experiment on the presented dataset in the ClowdFlows environment employs processing components (widgets) in a complex data analysis workflow which is shown in Fig. 4.

In the first step, the *Load Dataset* widget is used to read the breast cancer patient data, i.e., a binarized version of the gene expression data (note that the frozen robust multiarray analysis (fRMA) normalization (McCall et al. 2010) is also available from the InSilico web page). As the GSE2990 dataset does not have pre-specified classes we have selected the *Grade* attribute as the target attribute using the *Select Attributes* widget. According to Elston and Ellis (1991) and Galea et al. (1992), histologic grade of breast carcinomas provides clinically important prognostic information. Approximately one half of all breast cancers are assigned histologic grade 1 or 3 status (low or high risk of recurrence) but a substantial percentage of tumors (30–60 %) are classified as histologic grade 2 (intermediate risk of recurrence) which is not informative for clinical decision making

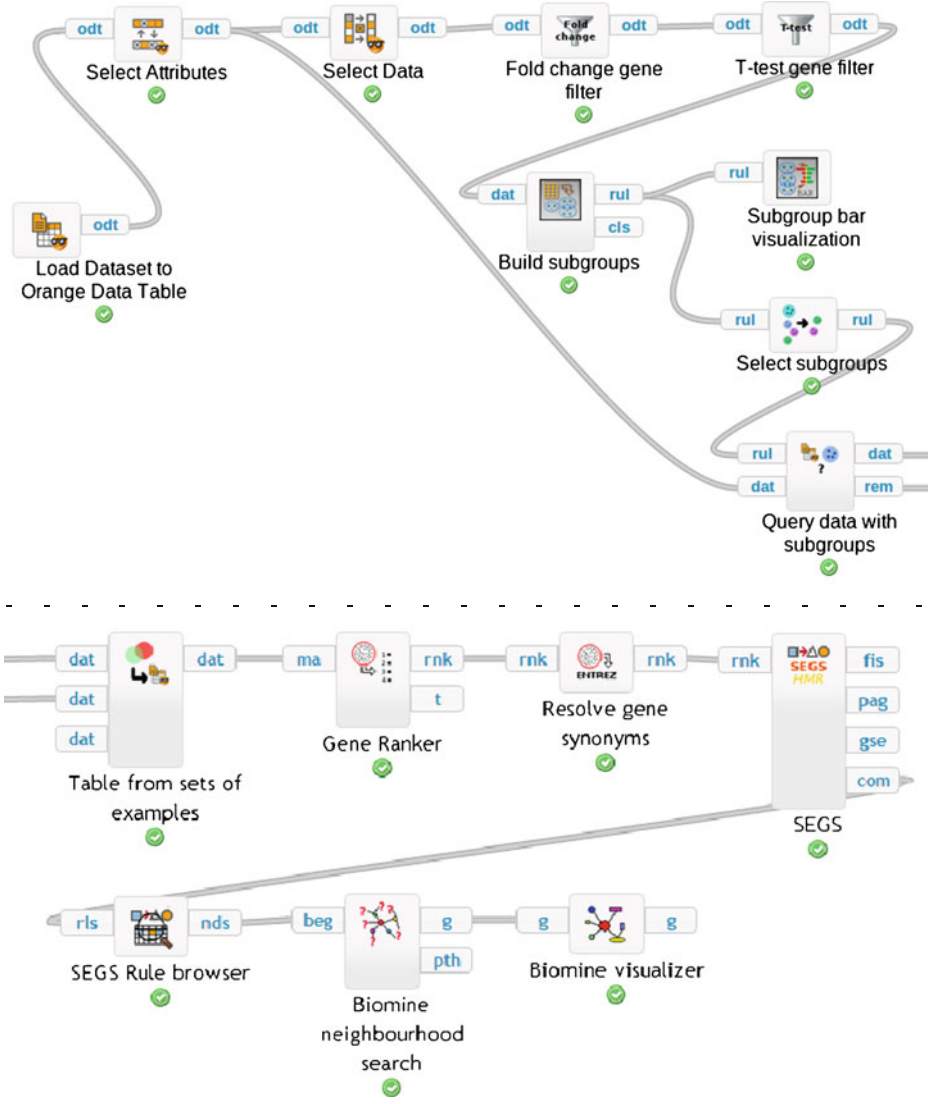


Fig. 4 A workflow implementing the proposed methodology in ClowdFlows (*first part*). The workflow was split into two parts in order to be more easily readable. The workflow can be found at http://clowdflows.org/workflow/911/

(Sotiriou et al. 2006). Obviously, to increase the prognostic value of tumor grading, further refinement of histologic grade 2 status is necessary (Sotiriou et al. 2006).

The third step of the workflow is to use the *Select data* widget to remove 17 unclassified examples for which the histologic grade is unknown. Although these examples may contain important information, this would require using unsupervised methods (e.g. clustering) instead of supervised subgroup discovery algorithms used in our experiments (note, however, that subgroup discovery in the presented workflow can easily be replaced by clustering or some other unsupervised method).

Next, attribute (gene) selection is performed using two gene filtering components which allow filtering the genes according to two scoring methods: fold change and t-test. Removal of unimportant genes is needed to reduce the search space of subgroup discovery methods to avoid the high-dimensionality problem. In our approach we have selected the genes in two stages: first, only the genes with a fold change of > 1 are selected, and second, only the genes with p-value $< 0,01$ given by the t-test are selected. This yields a total of 399 genes to be used in the subgroup discovery process.

The *Build subgroups* widget implements SD (Gamberger and Lavrač 2002), APRIORI-SD (Kavšek and Lavrač 2006) and CN2-SD (Lavrač et al. 2004) subgroup discovery algorithms while the *Subgroup Bar visualization* component provides a facility of bar chart visualization, while the *Select subgroups* widget allows the selection of particular subgroups. The selected subgroups are used to query the original data (*Query data with subgroups*) to obtain the covered set of examples which are then merged with the rest of the data (*Table from sets of examples*). As a result it is possible to rank the genes in the re-constructed dataset according to their ability to differentiate between the discovered subgroups and the rest of the data. The ranking of genes is performed by the *Gene ranker* widget implementing the ReliefF algorithm.

Finally, the computed ranking is sent to the *SEGS* widget which calls the web service implementing the SEGS semantic subgroup discovery algorithm (SDM-SEGS and SDM-Aleph can also be used). As the SEGS algorithm has large time and space requirements it is implemented as a web service which allows it to run on a powerful server. SEGS induces rules providing explanations of the top ranked attributes by building conjunctions of ontology terms from the GO ontology, KEGG orthology, and interacting terms using the Entrez gene-gene interactions database as described in Section 3.3. In our experiments we have used the latest updates of the ontologies and annotations provided by NCBI[6] and the Gene Ontology project.

The subgroup discovery analysis yielded two large subgroups (Table 2) of Grade 3 patients. Using the GeneCards[7] on-line tool, we have confirmed that all of the genes from the subgroup descriptions are typically differentially expressed (up-regulated) in breast cancer tissue when compared with normal tissue.

In the rest of this section we focus on the larger subgroup #1, for which we have generated explanations (Table 3). A total of 90 explanations with p-value < 0.05 (estimated using permutation testing) were found. Due to space restrictions we display only the top 10 explanations generated by SEGS (for a complete list open the workflow from Fig. 4). For example, Explanation #1 describes genes which are annotated by GO/KEGG terms: *chromosome* and *cell cycle*.

In the study by Sotiriou et al. (2006) where the expression profiles of Grade 3 and Grade 1 patients were compared, the genes that are associated with histologic grade were shown to be mainly involved in cell cycle regulation and proliferation (uncontrollable division of cells is one of the hallmarks of cancer). The explanations of Subgroup #1 of Grade 3 patients in Table 3 agree with their findings. In general, the explanations describe genes that take part in cell cycle regulation (Explanations #1–#10), cell division (Explanation #3) and other components that indirectly affect cell division (e.g., Explanations #4 and #5: microtubules are structures that pull the cell apart when it divides).

[6] http://www.ncbi.nlm.nih.gov/gene

[7] http://www.genecards.org

Table 2 The best-scoring subgroups found using CN2-SD with default parameters for the Grade 3 patients

#	Subgroup description	TP	FP
1	Grade = 3 ← DDX39A = 1 ∧ DDX47 = 1 ∧ RACGAP1 = 1 ∧ ZWINT = 1 ∧ PITPNB = 1	43	5
2	Grade = 3 ← TPX2 = 1 ∧ DDX47 = 1 ∧ PITPNB = 1 ∧ HN1 = 1	26	0

TP and FP are the true positive and false positive rates, respectively

In our implementation, the user can choose one of the explanations (i.e., gene sets) to query the Biomine database (Eronen and Toivonen 2012). The Biomine engine offers advanced probabilistic graph searching techniques that can be used to find a neighborhood of the set of genes, or a graph connecting two separate gene sets. The result of both is a subgraph that can be explored with the *Biomine visualizer* widget. Figure 5 shows a part of the neighborhood graph for the gene set of explanation #2. In this particular case, the figure shows three types of nodes (*gene*, *biological process* and *pathway*) and the links between the nodes signify how are the nodes related (*participates in*, *codes for*).

To sum up, our study shows that by using our methodology one can automatically reproduce the observations noted in the earlier work by Sotiriou et al. This can encourage the researchers to apply the presented methodology in similar exploratory analytics tasks. Given the easy access and adaptability of the software the methodology can be simply reused in other domains, which is demonstrated in the next section on financial news articles.

5.3 Supporting factors comparison

In this subsection we present the results of the related *supporting factors* (Gamberger and Lavrač 2003) methodology, which is also used to characterize subgroups and can be of great help to domain experts in the interpretation of subgroup discovery patterns. Supporting factors are features that have statistically significantly different distributions in the positive examples of a selected subgroup, when compared to the control examples (negative cases) in the whole population and by themselves do not appear in the subgroup description. The difference is measured using the χ^2-test of independence.

Table 3 The explanations for the patients from subgroup #1 from Fig. 3

#	Explanation	*p*-value
1	chromosome ∧ cell cycle	0.000
2	cellular macromolecule metabolic process ∧ intracellular non-membrane-bounded organelle ∧ cell cycle	0.000
3	cell division ∧ nucleus ∧ cell cycle	0.000
4	regulation of mitotic cell cycle ∧ cytoskeletal part	0.000
5	regulation of mitotic cell cycle ∧ microtubule cytoskeleton	0.000
6	regulation of G2/M transition of mitotic cell cycle	0.000
7	regulation of cell cycle process ∧ chromosomal part	0.000
8	regulation of cell cycle process ∧ spindle	0.000
9	enzyme binding ∧ regulation of cell cycle process ∧ intracellular non-membrane-bounded organelle	0.000
10	ATP binding ∧ mitotic cell cycle ∧ nucleus	0.005

We omit the variables from the rules for better readability. Note that since the *p*-values are estimations, some can also have a value of 0

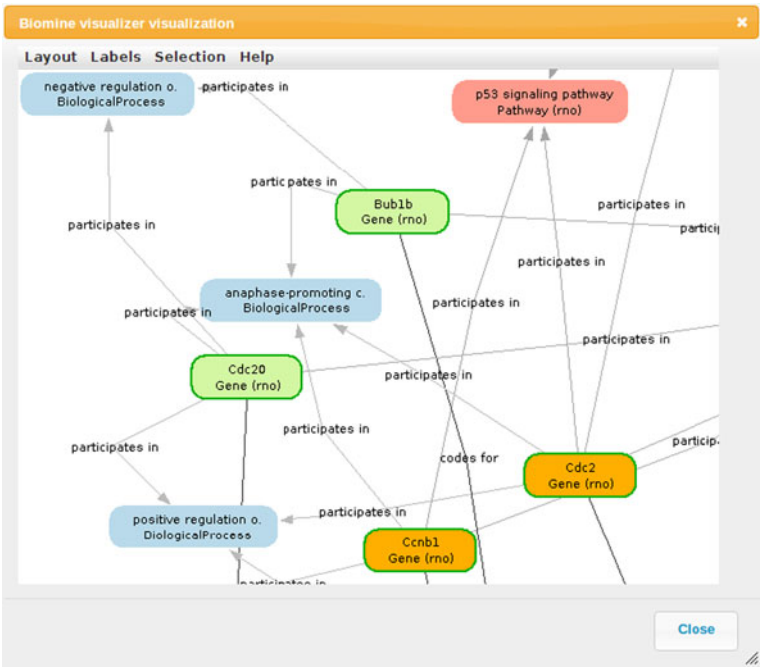


Fig. 5 The Biomine visualizer showing a part of a neighborhood graph for explanation #2 from Table 3

The methodology proposed in the present paper consists of several steps, where each step can be executed with multiple alternatives. The supporting factors methodology fits the best as a replacement to the last step. In this experiment, we assumed that all but the last step—explanation of subgroups—is the same as with our approach.

To be directly comparable to our results, we selected subgroup #1 (Fig. 2) as the target subgroup to characterize using supporting factors. We used a confidence level of 99 % ($p = 0.01$) and we report the best 10 supporting factors (Table 4).

The main differnce that we can see is that supporting factors are represented as single genes, and the technique does not try to generalize over the genes and associate them with concepts from the gene ontology. This can of course be desirable for many use cases, but in this genomics experiment the characterization is not instantly obvious, since an additional look-up of individual genes is required by the domain expert.

On the other hand, the reported genes reaffirm the higher-level explanations produced by our methodology. For example, again using the GeneCards tool, we can find that the TPX2 gene is required for normal assembly of microtubules during apoptosis (cell death).

Table 4 Subgroup #1 from Fig. 2 and its top 10 supporting factors calculated with a confidence value of 99 % ($p = 0.01$)

Subgroup description	Supporting factors
Grade $= 3 \leftarrow$ DDX39A $= 1 \wedge$ DDX47 $= 1$ $\wedge$RACGAP1 $= 1 \wedge$ ZWINT $= 1 \wedge$ PITPNB $= 1$	TPX2, MAD2L1, CCNB2, CDK1, NUSAP1, CENPA, SNRPD1, GINS1, ASPM, PRC1

CCNB2 plays a key role in the control of the cell cycle and NUSAP1 is another microtubule-associated protein. The GINS1 plays an essential role in the initiation of DNA replication.

To sum up, the supporting factors approach can be important in domains where extra supportive evidence is needed (e.g., medical decision support), since it lists specific features that support a given subgroup. On the other hand, it does not provide a more general context, such as is possible using semantic subgroup discovery methods. Of course, the expert could also benefit from using these two methodologies side-by-side, since they characterize subgroups at two different levels of abstraction.

6 Conclusions

In this paper we presented a methodology for explaining subgroups or sets of instances using higher-level ontological concepts. First, a subgroup of instances is identified (e.g., using subgroup discovery or clustering), which is then characterized using ontological concepts thus providing insight into the main differences between the given subgroup and the remaining data.

We made the developed tools available for the ClowdFlows platform. Due to this implementation the tools are easily accessible, since ClowdFlows requires only an internet connection and a web browser.

As demonstrated by the two use cases, the proposed approach is general and can be employed in any application area, provided the existence of available domain ontologies and annotated data to be analyzed. In this paper, the real-life use case is from the genomics domain.

As the experts assume that there are several molecular subtypes of breast cancer, our main research interest of the genomics use case is to employ the presented methodology to descriptively characterize the hypothesized cancer subtypes. Hence, in addition to distinguishing between patients with breast cancer (the positive cases) and healthy patients, the challenge is to identify breast cancer subtypes by finding subgroups of patients which would be explained by the same gene functions, processes in which the genes interact. The approach presented in this paper has the potential of discovering groups of patients which correspond to the subtypes while explaining them using ontology terms describing gene functions, processes and pathways; in this paper, we applied the methodology to describe breast cancer grades with the aim of evaluation.

Using subgroup discovery we have identified two main subgroups that characterize Grade 3 breast cancer patients. These were then additionally explained using Gene Ontology concepts and KEGG pathways and the explanations (rules or subgroup descriptions of gene sets) agree with previous findings characterizing grades using microarray profiling.

Furthermore, compared to the related supporting factors approach, which is also used to characterize subgroups, the experiments show that supporting factors are more useful when concentrating on specific low-level attributes or features is desirable, but when more general descriptions are needed, they are not as easy to interpret as the method presented in this paper. This restriction is even more apparent on gene expression data, since the supporting factors are given in terms of single genes.

The results of the conducted experiments show the capabilities of the presented approach. In further work we will employ the methodology to detecting and characterizing subtypes of breast cancer. In further work, we will apply this methodology to other domains, as well as advance the level of exploitation of domain ontologies for providing explanations of the results of data mining.

Acknowledgments This work was supported by the Slovenian Ministry of Higher Education, Science and Technology [grant number P-103], the Slovenian Research Agency [grant number PR-04431], the SemDM project (Development and application of new semantic data mining methods in life sciences) [grant number J2-5478] and the FP7 European Commission project MUSE (Machine understanding for interactive storytelling) [grant number 296703].

References

Angiulli, F., Fassetti, F., Palopoli, L. (2013). Discovering characterizations of the behavior of anomalous subpopulations. *IEEE Transactions on Knowledge and Data Engineering*, *25*(6), 1280–1292. doi:10.1109/TKDE.2012.58.

Atzmüller, M., & Puppe, F. (2006). SD-Map—a fast algorithm for exhaustive subgroup discovery. In *Proceedings of the 10th European conference on principles and practice of knowledge discovery in databases (PKDD '06)* (pp. 6–17). Springer.

Bay, S.D., & Pazzani, M.J. (2001). Detecting group differences: mining contrast sets. *Data Mining and Knowledge Discovery*, *5*(3), 213–246.

Demšar, J., Zupan, B., Leban, G. (2004). *Orange: from experimental machine learning to interactive data mining, white paper*. Faculty of Computer and Information Science, University of Ljubljana. www.ailab.si/orange.

Dong, G., & Li, J. (1999). Efficient mining of emerging patterns: discovering trends and differences. In *Proceedings of the 5th ACM SIGKDD international conference on knowledge discovery and data mining (KDD-99)* (pp. 43–52).

Elston, C.W., & Ellis, I.O. (1991). Pathological prognostic factors in breast cancer. I. The value of histological grade in breast cancer: experience from a large study with long-term follow-up. *Histopathology*, *19*(5), 403–410.

Eronen, L., & Toivonen, H. (2012). Biomine: predicting links between biological entities using network models of heterogeneous databases. *BMC Bioinformatics*, *13*, 119.

Galea, M., Blamey, R., Elston, C., Ellis, I. (1992). The Nottingham prognostic index in primary breast cancer. *Breast Cancer Research and Treatment*, *22*, 207–219.

Gamberger, D., & Lavrač, N. (2002). Expert-guided subgroup discovery: methodology and application. *Journal of Artificial Intelligence Research (JAIR)*, *17*, 501–527.

Gamberger, D., & Lavrač, N. (2003). Active subgroup mining: a case study in coronary heart disease risk group detection. *Artificial Intelligence in Medicine*, *28*(1), 27–57.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H. (2009). The WEKA data mining software: an update. *SIGKDD Explor Newsl*, *11*, 10–18.

Hilario, M., Nguyen, P., Do, H., Woznica, A., Kalousis, A. (2011). Ontology-based meta-mining of knowledge discovery workflows. In N. Jankowski, W. Duch, K. Grabczewski (Eds.), *Meta-learning in computational intelligence, studies in computational intelligence* (Vol. 358, pp. 273–315). Berlin Heidelberg: Springer.

Jovanoski, V., & Lavrač, N. (2001). Classification rule learning with APRIORI-C. In P. Brazdil & A. Jorge (Eds.), *EPIA, lecture notes in computer science* (Vol. 2258, pp. 44–51). Berlin Heidelberg: Springer.

Kavšek, B., & Lavrač, N. (2006). APRIORI-SD: adapting association rule learning to subgroup discovery. *Applied Artificial Intelligence*, *20*(7), 543–583.

Klösgen, W. (1996). Explora: a multipattern and multistrategy discovery assistant. In *Advances in knowledge discovery and data mining*, (pp. 249–271). Menlo Park: American Association for Artificial Intelligence.

Kralj Novak, P., Lavrač, N., Webb, G.I. (2009). Supervised descriptive rule discovery: a unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research*, *10*, 377–403.

Kranjc, J., Podpečan, V., Lavrač, N. (2012). Clowdflows: a cloud based scientific workflow platform. In P.A. Flach, T.D. Bie, N. Cristianini (Eds.), *ECML/PKDD (2), lecture notes in computer science* (Vol. 7524, pp. 816–819). Berlin Heidelberg: Springer.

Langohr, L., Podpečan, V., Petek, M., Mozetič, I., Gruden, K., Lavrač, N., Toivonen, H. (2013). Contrasting subgroup discovery. *Computer Journal*, *56*(3), 289–303.

Lavrač, N., Kavšek, B., Flach, P.A., Todorovski, L. (2004). Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, *5*, 153–188.

Lavrač, N., Vavpetič, A., Soldatova, L., Trajkovski, I., Kralj Novak, P. (2011). Using ontologies in semantic data mining with SEGS and g-SEGS. In *Proceedings of the international conference on discovery science* (*DS '11*) (pp. 165–178). Springer.

Lawrynowicz, A., & Potoniec, J. (2011). Fr-ont: an algorithm for frequent concept mining with formal ontologies. In M. Kryszkiewicz, H. Rybinski, A. Skowron, Z.W. Ras (Eds.), *ISMIS, lecture notes in computer science* (Vol. 6804, pp. 428–437). Berlin Heidelberg: Springer.

Maglott, D., Ostell, J., Pruitt, K.D., Tatusova, T. (2005). Entrez gene: gene-centered information at NCBI. *Nucleic Acids Research*, *33*(Database issue).

McCall, M.N., Bolstad, B.M., Irizarry, R.A. (2010). Frozen robust multiarray analysis (fRMA). *Biostatistics*, *11*(2), 242–253.

Podpečan, V., Juršič, M., Žakova, M., Lavrač, N. (2009). Towards a service-oriented knowledge discovery platform. In V. Podpečan & N. Lavrač (Eds.), *Third-generation data mining: towards service-oriented knowledge discovery* (pp. 25–36).

Podpečan, V., Lavrač, N., Mozetič, I., Kralj Novak, P., Trajkovski, I., Langohr, L., Kulovesi, K., Toivonen, H., Petek, M., Motaln, H., Gruden, K. (2011a). SegMine workflows for semantic microarray data analysis in Orange4WS. *BMC Bioinformatics*, *12*, 416.

Podpečan, V., Zemenova, M., Lavrač, N. (2011b). Orange4WS environment for service-oriented data mining. *The Computer Journal*. doi:10.1093/comjnl/bxr077. Accessed 7 Aug 2011.

Robnik-Šikonja, M., & Kononenko, I. (2003). Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, *53*, 23–69.

Sotiriou, C., Wirapati, P., Loi, S., Harris, A., Fox, S., Smeds, J., Nordgren, H., Farmer, P., Praz, V., Haibe-Kains, B., Desmedt, C., Larsimont, D., Cardoso, F., Peterse, H., Nuyten, D., Buyse, M., Van de Vijver, M.J., Bergh, J., Piccart, M., Delorenzi, M. (2006). Gene expression profiling in breast cancer: understanding the molecular basis of histologic grade to improve prognosis. *Journal of the National Cancer Institute*, *98*(4), 262–272.

Srinivasan, A. (2007). *Aleph manual*. http://www.cs.ox.ac.uk/activities/machinelearning/Aleph/.

Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Paulovich, A., Pomeroy, S.L., Golub, T.R., Lander, E.S., Mesirov, J.P. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(43), 15,545–15,550.

Suzuki, E. (1997). Autonomous discovery of reliable exception rules. In *Proceedings of the third international conference on knowledge discovery and data mining* (pp. 259–262).

Suzuki, E. (2006). Data mining methods for discovering interesting exceptions from an unsupervised table. *Journal of Universal Computer Science*, *12*(6), 627–653.

Taminau, J., Steenhoff, D., Coletta, A., Meganck, S., Lazar, C., de Schaetzen, V., Duque, R., Molter, C., Bersini, H., Nowé, A., Weiss Solís, D.Y. (2011). InSilicoDB: an R/Bioconductor package for accessing human Affymetrix expert-curated datasets from GEO. *Bioinformatics*. doi:10.1093/bioinformatics/btr529.

Trajkovski, I., Lavrač, N., Tolar, J. (2008). SEGS: search for enriched gene sets in microarray data. *Journal of Biomedical Informatics*, *41*(4), 588–601.

Vavpetič, A., & Lavrač, N. (2013). Semantic subgroup discovery systems and workflows in the SDM-Toolkit. *Computer Journal*, *56*(3), 304–320.

Vavpetič, A., Podpečan, V., Meganck, S., Lavrač, N. (2012). Explaining subgroups through ontologies. In P. Anthony, M. Ishizuka, D. Lukose (Eds.), *Proceedings of PRICAI, lecture notes in computer science* (Vol. 7458, pp. 625–636). Berlin Heidelberg: Springer.

Vavpetič, A., Novak, P.K., Grčar, M., Mozetič, I., Lavrač, N. (2013). Semantic data mining of financial news articles. In *Proceedings of the international conference on discovery science (DS '13)*. Springer.

Webb, G.I., Butler, S.M., Newlands, D. (2003). On detecting differences between groups. In *Proceedings of the 9th ACM SIGKDD international conference on knowledge discovery and data mining (KDD-03)* (pp. 256–265).

Wrobel, S. (1997). An algorithm for multi-relational discovery of subgroups. In *Proceedings of the first European conference on principles of data mining and knowledge discovery (PKDD '97)* (pp. 78–87). Springer.

Žáková, M., Železný, F., García-Sedano, J.A., Tissot, C.M., Lavrač, N., Kremen, P., Molina, J. (2006). Relational data mining applied to virtual engineering of product designs. In *Proceedings of the 16th international conference on inductive logic programming (ILP'06)* (pp. 439–453). Berlin/Heidelberg, Germany, Santiago de Compostela, Spain: Springer-Verlag.

5.2 Multi-resolution 0–1 Data Analysis

This section describes a methodology for characterizing data clusters, where semantic subgroup discovery is one of the main steps, together with visualization using banded matrices.

5.2.1 Methodology

The proposed three-part methodology for multiresolution 0–1 data analysis consists of data clustering with mixture models, extraction of rules from clusters, as well as data and rule visualization using banded matrices. The results of the three-part process—clusters, rules from clusters, and banded structure of the data matrix—are finally merged in a unified visual banded matrix display. The incorporation of multiresolution data is enabled by the supporting ontology, describing the relationships between the different resolutions.

5.2.2 Experimental results

The methodology was applied to a multiresolution chromosomal amplification dataset and to four non-biomedical datasets. The dataset describes DNA copy number amplifications in 4,590 cancer patients. The data describes 4,590 patients as data instances, with attributes being chromosomal locations indicating amplifications in the genome. These aberrations are described as 1's (amplification) and 0's (no amplification). Amplification data is further described at two different resolution levels (312 and 393 locations, for 24 different chromosomes).

We focused on 34 (out of 71) most frequent cancers—covering 90% of the data—and on one chromosome. In addition, we used supplementary background knowledge in the form of an ontology. This consisted of the hierarchical structure of multiresolution amplification data, chromosomal locations of fragile sites, virus integration sites, cancer genes, and amplification hot spots.

In addition to the chromosomal amplifications data, we tested our methodology on four publicly available data sets originally used in [76].

1. **Cities** data set describes the most and least liveable cities in the world according to the Mercer ranking.

2. **NY Daily** data set describes the crawled news items along with their sentiment scores.

3. **Tweets** data set is a collection of tweets with different features where the original task is to identify different sports related tweets.

4. **Stumble Upon** data set consists of training data set used in the Kaggle competition.

To generate the hierarchical features, the 'DBpedia Direct Types' ontology was used in the first three experiments, and the 'Open directory project' ontology was used to extract categories for each URL in the fourth data set, i.e. we used the same approach as in the original experiments reported in [76].

In summary, the three-part process together offers an improved view of the structure of the underlying data. The visualizations show that most of the samples in the same cluster also come together in the banded matrix visualization. This has been achieved by reordering the matrix rows by placing similar items closer together to form a banded structure, which allows easier visualization of the clusters and rules.

The resulting rules offer short and precise descriptions of the clusters; in particular cases also due to the additional hierarchical background knowledge. Furthermore, the super-positioning of rules onto the banded matrix visualization offers a unique perspective on the rules as well as on the clusters. It clearly visualizes which features are discriminative and which are not, and why. An extensive description of the work is available in the following journal publication.

5.2.3 Related publication

Details of the methodology and experiments can be found in the following journal article (included in this section):

P. R. Adhikari, A. Vavpetič, J. Kralj, N. Lavrač, and J. Hollmén, "Explaining mixture models through semantic pattern mining and banded matrix visualization," *Machine Learning Journal*, in press 2016.

The author's contributions are as follows. Prem Raj Adhikari contributed the mixture modelling part of the experiment and expert analysis of the results. Anže Vavpetič contributed the semantic pattern discovery part of the experiment. Jan Kralj contributed the banded matrices and visualization parts of the experiment. Nada Lavrač and Jaakko Hollmén contributed the idea of the three-part methodology. All authors contributed equally to the methodology and experimental design, and to the text of the publication.

Machine Learning manuscript No.
(will be inserted by the editor)

Explaining mixture models through semantic pattern mining and banded matrix visualization

Prem Raj Adhikari · Anže Vavpetič · Jan Kralj · Nada Lavrač · Jaakko Hollmén

the date of receipt and acceptance should be inserted later

Abstract This paper presents an approach to semi-automated data analysis, supported by tools for pattern construction, exploration and explanation. The proposed three-part methodology for multiresolution 0-1 data analysis consists of data clustering with mixture models, extraction of rules from clusters, as well as data and rule visualization using banded matrices. The results of the three-part process: clusters, rules from clusters, and banded structure of the data matrix are finally merged in a unified visual banded matrix display. The incorporation of multiresolution data is enabled by the supporting ontology, describing the relationships between the different resolutions, which is used as background knowledge in the semantic pattern mining process of descriptive rule induction. The presented experimental use case highlights the usefulness of the proposed methodology for analyzing complex DNA copy number amplification data, studied in previous research, for which we provide new insights in terms of induced semantic patterns and cluster/pattern visualization. The methodology is successfully evaluated on four other publicly available data sets, which further demonstrates the utility of the proposed approach.

Keywords Mixture Models · Clustering · Semantic Pattern Mining · Banded Matrix · Pattern Visualization

Prem Raj Adhikari · Jaakko Hollmén
HIIT and Department and Computer Science, Aalto University School of Science
PO Box 15400, FI-00076 Aalto, Espoo, Finland
E-mail: prem.adhikari@utu.fi, jaakko.hollmen@aalto.fi
Present affiliation of first author: Department of Physiology and Turku Center for Disease Modeling, Institute of Biomedicine, University of Turku, Turku, Finland

Anže Vavpetič · Jan Kralj · Nada Lavrač
Jožef Stefan Institute and Jožef Stefan International Postgraduate School
Jamova 39, 1000 Ljubljana, Slovenia
E-mail: {anze.vavpetic, jan.kralj, nada.lavrac}@ijs.si

1 Introduction

Data analysis is concerned with finding ways to summarize the data to become easily understandable [22]. The interpretation aspect is especially valued among domain specialists who may not understand the data analysis process itself. In the current age of big data and accompanying complex models, understandability and interpretability of the models is even more essential as, according to Richard Hamming, "the purpose of computing is insight, not numbers" [21]. For complex models generated from big data, understanding these models will help understanding the data generating phenomenon and help making better decisions based on the data [37]. Semi-automated data analysis is hence made possible for the end-user if data analysis processes are supported by easily accessible methodologies and tools for pattern and model construction, as well as their exploration and explanation.

This work combines different approaches developed in our previous research, leading to a new three-part data analysis methodology, whose utility is demonstrated in a case study concerning the analysis of DNA copy number amplifications represented as a 0-1 (binary) data set [51]. In previous work we have successfully clustered this data using mixture models [52, 67]. Furthermore, in [26], we have learned linguistic names for the patterns that coincide with the natural structure in the data, enabling domain experts to use these names to refer to the clusters or to the patterns extracted from the clusters. In [25] we reported that frequent itemsets describing the clusters, or extracted from the 'one cluster at a time' clustered data differ from those extracted from the whole data set. The whole set of about 100 DNA amplification patterns identified from the data have been described in [52].

In the proposed approach we start from our initial studies of using mixture models to crossover unsupervised methods of probabilistic clustering with supervised methods of subgroup discovery with the aim to determine the chromosomal locations that are responsible for specific types of cancers. We also enrich the data with additional background knowledge that enables the analysis of data at multiple resolution levels. Specifically, with the aim of better explaining the initial mixture model based clusters, the proposed methodology considers the cluster identifiers as class labels for descriptive rule learning [53], using semantic pattern mining [74]. The resulting semantic rules are generated by the Hedwig semantic pattern mining algorithm [73] performing semantic subgroup discovery by using the incorporated background knowledge in the form of pre-discovered patterns as well as taxonomies of features in multiresolution data. Finally, we use a banded matrix approach to visualize the clustering result and rules obtained from semantic subgroup discovery overlayed on the same data, thus providing holistic picture of the data and consequently, of the data generating phenomenon.

Explaining the obtained clustering results to the users is essential. It was shown that in text mining [28], semantic structures can be used to explain the clustering results at an appropriate level of granularity. Similarly, a methodology consisting of clustering and semantic pattern mining, has already been

suggested in our previous work [38, 74]. However, in this work we have for the first time addressed the task of explaining sub-symbolic mixture model patterns (clusters of instances) using symbolic rules. To this end, we propose our previous approach [74] to be enhanced through pattern comparison by their visualization on the plots resulting from banded matrices visualization [20]. Using different color schemes on the banded matrix structure (induced from the original data), the mixture model clusters are first visualized, followed by visualizing the sets of patterns (i.e. subgroups) induced by semantic pattern mining. The proposed visualization provides new means for data and pattern exploration and comparison. To the best of our knowledge, such a three-part exploratory approach to data analysis has not been proposed in the data mining literature before.

The main contribution of this work is a three-part methodology for data analysis, consisting of (i) data clustering, (ii) extraction of semantic patterns (rules) from the clusters, using an ontology of relationships between the different resolutions of the multiresolution data, and (iii) integration of the results in a visual display, illustrating the clusters and the identified rules by visualizing them over the banded matrix structure, first described in [3]. This work significantly extends our previous report on the same topic [3] in many ways. First of all, we used a more elaborate experimental setting with four additional data sets. Furthermore, we added a new section on literature survey where we present the state-of-the-art in all three methodological parts in our contribution as well as the holistic picture of similar methodologies, and sections detailing the model selection procedure in mixture models and performed statistical tests for empirical verification of stability of the clustering results. We also changed a part of the methodology, replacing one banded matrix alorithm (the barycentric method) with another (the bidirectional MBA) which yielded better results in our experiments.

The paper is structured as follows. The related work is presented in Section 2. Methodology overview along with the details are explained in Section 3. Section 4 describes the experimental data sets. Section 5 describes the experiments on the chromosomal amplification data set and their results, while Section 6 presents the experiments on four additional publicly available data sets. We present the results of the stability analysis of clustering results in Section 7. In Section 8, we summarize the results and conclude the paper.

2 Related work

The following sections provide a brief overview of related work in mixture modeling, analysis of multiresolution data, semantic pattern mining, and pattern visualization using banded matrices. In the end of this section, we review some of the research that investigates at least two aspects of our three-part methodology.

2.1 Mixture models

Mixture models have been popular in the probabilistic modeling domain because of their flexibility in the choice of component distributions and their applicability to a wide variety of applications. Mixture models are at the heart of model based clustering [49]. Authors in [49] review the model based clustering approach in different application areas such as text mining, proteomics, and medical data analysis. Similarly, authors in [47] summarize different application areas where mixture models have been used with plausible results such as density estimation, missing data imputation, combining different density models, and model heterogeneity. In our earlier work, mixture models were used to model heterogeneous cancer patient data [52, 67].

2.2 Mixture models in copy number analysis

In the beginning, DNA copy number analysis focused in determining the copy number of the cytogenetic bands [35, 59]. However, in [35] and [59] the authors did not establish a relation between the copy numbers and their clinical significance.

DNA copy number amplification data collected from bibliomics survey from 838 journal articles published from 1992 to 2002 was analyzed in [51], where amplification patterns were determined for 73 different neoplasms and the neoplasms were clustered according to amplification profiles thus identifying the amplification hotspots using independent component analysis. The profiling revealed that human neoplasms formed clusters based on the amplification frequency of the cancer. Similarly, authors in [52] classified the human cancers based on copy number amplification using probabilistic modeling. Furthermore, the authors extracted the ranges of the amplification in the chromosome and expressed it according to the cytogentic nomenclature.

In [26] and [67], the authors modeled the DNA copy number amplifications using a mixture of multivariate Bernoulli Distributions. The classification of 73 different neoplasms in [51] were extended to 95 different neoplasm types. Furthermore, in [60], the authors have proposed the enhancement to Bayesian Piecewise Constant Regression (BPCR), called mBPCR, changing the segment number estimator and boundary estimator to enhance the fitting procedure. The proposed mBPCR was more accurate in the determination of true breakpoints of amplification. More recent studies [14] and [15] have mainly focused in cancer specific analysis of DNA copy number.

2.3 Multiresolution data analysis

Multiresolution data arises when a phenomenon is measured with varying precision [78]. A phenomenon measured with increasing precision measures the finer details of the phenomenon and produces the data in fine resolution.

In contrast, a phenomenon measured with decreasing precision measures the coarser details of the phenomenon and produces data in coarse resolution. Multiresolution data are abundant in domains such as time series, image processing, geoinformatics, and telecommunications [78]. Multiresolution methods are gaining popularity in recent years because of their ability to model data in multiple dimensions within a single analysis, providing means to combine multiple data sets and sources within a single analysis framework.

Multiresolution modeling is closely related to the scale space theory [42] and multiscale analysis [77] and the terms are sometimes used interchangeably in the literature. Multiscale representation is often generated from single resolution data by successive smoothing and subsampling, for example, by using the pyramid structure in image processing domain [42]. Scale space representation improves over multiscale representation by providing facilities to compute representation using a desired scale parameter, t. Scale space and multiscale methods work in the model domain where models represent single resolution data at different scales. In contrast, multiresolution modeling problem arises in the data domain where the same data generating system is measured at varying levels of detail. Wavelets describe mathematical phenomena such as functions and signals at different levels of resolution but in a regular, consistent and homogeneous setting [32]. Most of propositional machine learning and data mining methods described in the literature are designed to work with single resolution data. Since the dimensionality of different data resolutions is different, the usual approach is to model each resolution separately. Scale space methods and wavelets usually use a multiresolution analysis setting for the data sets in the same resolution. Furthermore, the multiresolution scenarios where wavelets and scale space methods have their usage require regular, consistent, and homogeneous division of regions such as the pyramid structure in the image processing domain [79]. In a multiresolution setting, the division is consistent but irregular because a region in a coarse resolution is not always divided into the same number of regions in a fine resolution like in our multiresolution chromosomal amplification data sets.

Multiresolution mixture models have been proposed in the literature. For example, a multiresolution Gaussian mixture model founded on the pyramid structure in image processing domain models the visual motion in [79]. Authors in [50] incorporate wavelet sub-bands in a Gaussian mixture model to improve their performance thereby providing a generic platform to use any multiresolution decomposition based Gaussian mixture model for background suppression. We adapted mixture modeling for multiresolution data in our past research. In [1], we transformed the multiresolution data to a single resolution and applied the mixture modeling algorithm on the combined data thus increasing the performance of mixture models on single resolution data. In [2], we showed the improvement in the modeling performance of multiresolution mixture model by designing the structure of multiresolution components from the domain knowledge for the mixture model such that a single multiresolution component is a Bayesian network.

2.4 Semantic pattern mining

Rule learning, which was initially focused on building predictive models formed of sets of classification rules, has recently shifted its focus to descriptive pattern mining. Well-known pattern mining techniques are based on association rule learning [4, 58]. While the initial studies in association rule mining have focused on finding interesting patterns from large data sets in an unsupervised setting, association rules have been used also in a supervised setting, to learn pattern descriptions from class-labeled data [43]. Building on top of the research in classification and association rule learning, subgroup discovery has emerged as a popular data mining methodology for finding patterns in class-labeled data. Subgroup discovery aims at finding interesting patterns as individual rules that best describe the target variable [34, 81].

Subgroup descriptions in the form of propositional rules are suitable descriptions of groups of instances. However, given the abundance of taxonomies and ontologies that are readily available, these can also be used to provide higher-level descriptors and explanations of discovered subgroups. Especially in the domain of systems biology, the GO ontology [12], KEGG orthology [55] and Entrez gene–gene interaction data [44] are good examples of structured domain knowledge that can be used as additional higher-level descriptors in the induced rules.

The challenge of incorporating domain ontologies in data mining was addressed in recent research on semantic data mining (SDM) [41, 72]. Using ontologies, authors in [41] introduce an algorithm named Fr–ONT for frequent concept mining expressed in $\mathcal{EL}^{++}$ DL. In [72] we described and evaluated the SDM toolkit that includes two semantic data mining systems: SDM-SEGS and SDM-Aleph. SDM-SEGS is an extension of the earlier domain-specific algorithm SEGS [68] which allows for semantic subgroup discovery in gene expression data. SEGS constructs gene sets as combinations of GO ontology [12] terms, KEGG orthology [55] terms, and terms describing gene–gene interactions obtained from the Entrez database [44]. SDM-SEGS extends and generalizes this approach by allowing the user to input any set of ontologies in the OWL ontology specification language and an empirical data collection which is annotated by domain ontology terms. SDM-SEGS employs ontologies to constrain and guide the top-down search of a hierarchically structured space of induced hypotheses. SDM-Aleph, which is built using the inductive logic programming system Aleph [64], does not have the limitations of SDM-SEGS, imposed by the domain-specific algorithm SEGS. Additionally, SDM-Aleph can accept any number of OWL ontologies as background knowledge, which are then used in the learning process.

Based on the lessons learned in [72], we introduced a new system Hedwig in [73]. The system takes the best from both SDM-SEGS and SDM-Aleph. It uses an efficient search mechanism tailored to exploit the hierarchical nature of ontologies. Furthermore, Hedwig can take into account background knowledge in the form of RDF triplets. Compared to [73], we upgraded the original system to use better redundancy pruning and significance tests based on [30]. The

latest version of Hedwig supports also negations of unary predicates. This version of the Hedwig system was used in the experiments described in this paper.

2.5 Related methodologies

Complex models are needed for modeling complex, non-linear relationships in the data. As argued in [66], however, complex models exhibit a low degree of human comprehensibility. Rules can be used to represent complex models, since they have the advantage of being compact, modular, explicit and interpretable by domain experts [70]. In our current work, we use semantic pattern mining in order to represent the clustered data in an interpretable fashion. Another line of work is to summarize the clustered data in an interpretable fashion in the context of topic models [48, 39]. Having identified topics as clusters in a document collection, the task is to summarize the contents of that cluster or topic in a concise way.

Work presented in [70] considers relationships between probabilistic rules, normalized Gaussian basis functions and Gaussian mixture models, which can be seen as different representational forms of knowledge. The work considers extracting rules out of models, but also the use of rules to support model estimation. Rule extraction from feed-forward neural networks is investigated in [66]. In that work, rules are extracted, where the precondition is given by a set of intervals for the individual values and the output is a single target category.

The aim of the research presented in [39] is to automatically generate topic labels which explicitly identify the semantics of the topic. The work in [48] proposes probabilistic approaches to automatically labeling multinomial topic models in an objective way.

2.6 Data clustering and visualization using banded matrices

Data visualization has been an integral ingredient in the overall data mining process because it presents insights into complex data sets by communicating their key aspects [71]. Furthermore, providing information in the visual format is one of the fastest and best methods understandable to domain experts. Data is often represented in a matrix form, and research community has developed numerous methods for matrix visualization [9, 82]. In this contribution, we use banded matrices to visualize the data and the results of a data mining process in a way that the results become easily understandable to the domain specialist.

While binary matrices are frequently used as input in data mining (perhaps the most notable example of binary matrices being market basket data), the concept of banded matrices has its origins in numerical analysis. This is because the computational effort of multiplying matrices is much smaller when

matrices are banded. The interest of the numerical community is usually in reducing the total bandwidth of a matrix. This differs slightly from the interests in data mining, where the goal is to find a matrix structure as close to a banded one with the underlying assumption that the data analyzed is noisy and contains outliers. The connection between banded matrices and their relation to data analysis was initially studied in [20], where several algorithms were proposed to find optimal permutations of rows (and sometimes columns) that best expose the banded structure of a matrix.

In this work we conducted experiments with three algorithms: minimal banded augmentation (MBA), bidirectional MBA (biMBA), and the barycentric method. Given that the performance of the biMBA method, first proposed in [20], was superior to both MBA and the barycentric method, we used this method in the visualization.

3 Methodology

This section describes the proposed three-part methodology of our contribution. The three steps consist of clustering with mixture models, a subsequent cluster explanation through pattern construction using semantic pattern mining, and finally pattern visualization enabling improved pattern interpretation.

3.1 Methodology overview

The proposed methodology is illustrated in Figure 1. The input to the methodology pipeline is the experimental data and the background knowledge, which defines the taxonomy of attribute values at different levels of the given multiresolution data, with locations for various factors that are known to contribute to cancer development or are characteristic of most cancer types.

The first step in the methodology pipeline is mixture modeling, consisting of model selection to determine the number of mixture components and probabilistic clustering to generate the cluster labels from the data. In the next step, data is structured using a banded matrix approach. While the banded structure is induced from the data independently of cluster labels and the background knowledge, the obtained banded structure can be used also to support the visualization of the clusters obtained through mixture modeling. Next, the data (labeled by cluster labels obtained from mixture modeling) and the background knowledge are used as input to the Hedwig semantic pattern mining algorithm, to get the descriptions of data clusters in the form of logical rules, whose conditions include conjunctions of background knowledge concepts. Semantic pattern mining is the only modeling approach in the methodology that uses the background knowledge and facts. Finally, all three models (the mixture model, the banded matrix and the patterns) are joined to produce the final banded matrix-based visualization.

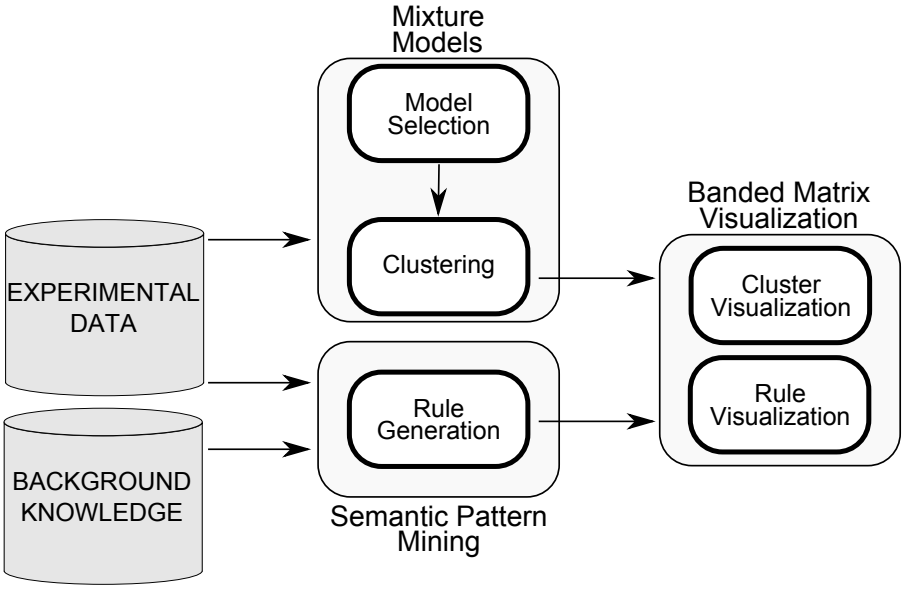


Fig. 1: Overview of the proposed three-part methodology used in the analysis of high-dimensional multiresolution data.

3.2 Mixture model clustering

Mixture models are probabilistic models for modeling complex distributions by a weighted sum, or a mixture of simple distributions. Mixture model decomposes the complex probability distribution into a set of component distributions [47]. The form of mixture distribution is dependent on the choice of the component distributions. Distributions from exponential family such as Gaussian and Dirichlet dominate the choice of component distributions [47]. Since the data set of our interest is a 0-1 data, we use multivariate Bernoulli distributions as component distributions to model the data. Mathematically, this can be expressed as:

$$P(\mathbf{x}) = \sum_{j=1}^{\mathrm{J}} \pi_j P(\mathbf{x} \mid \boldsymbol{\theta}_j) = \sum_{j=1}^{\mathrm{J}} \pi_j \prod_{i=1}^{\mathrm{d}} \theta_{ji}^{x_i} (1-\theta_{ji})^{1-x_i}. \tag{1}$$

Here, $j = 1, 2, \ldots, J$ indexes the component distributions and $i = 1, 2, \ldots, d$ indexes the dimensionality of the data. π_j defines the mixing proportions or mixing coefficients determining the weight for each of the J component distributions. The mixing coefficients satisfy the properties of convex combination, i.e. $\pi_j \geq 0$ and $\sum_{j=1}^{J} \pi_j = 1$. Individual parameters θ_{ji} determine the probability that a random variable in the j^{th} component in the i^{th} dimension takes the value 1. Parameters for a component distribution j is denoted as $\boldsymbol{\theta}_j = (\theta_{j1}, \theta_{j2}, \ldots, \theta_{jd})$. The term x_i denotes the data point such that

$x_i \in \{0, 1\}$, in the data vector $\mathbf{x} = (x_1, x_2, \ldots x_d)$. Therefore, the parameters of mixture models can be represented as: $\boldsymbol{\Theta} = \{J, \{\pi_j, \boldsymbol{\theta}_j\}_{j=1}^{J}\}$. We can formulate Equation 1 in log–likelihood terms according to maximum likelihood principle [7], where parameter values that maximise the log-likelihood can be defined as:

$$\mathcal{L}(\boldsymbol{\Theta}) = \sum_{n=1}^{\mathrm{N}} \log\ P(x_n \mid \boldsymbol{\Theta}) = \sum_{n=1}^{\mathrm{N}} \log \left[\sum_{j=1}^{\mathrm{J}} \pi_j \prod_{i=1}^{\mathrm{d}} \theta_{ji}^{x_{ni}} (1 - \theta_{ji})^{1 - x_{ni}} \right]. \quad (2)$$

3.2.1 Motivation behind using mixture models

Whereas the mixture model is merely a way to represent the probability distribution of the data, the model can be used in clustering the data into (hard) partitions, or subsets of data instances. We can achieve this by allocating individual data vectors to mixture model components that maximize the posterior probability of that data vector.

Among the diverse set of clustering methods of choice we chose mixture modeling because we wanted to model the data in a probabilistic context. Probabilistic models used in clustering provides several advantages over traditional clustering methods as they provide principled methods to address issues such as number of clusters, and missing variables [47]. Clustering methods such as k-means (which can also be interpreted as mixture models) use simple statistical measures such as mean, or median of data items in clusters, while we opted for mixture models that provide more complete information. When mixture models are used in clustering, the components represent the clusters thus making it possible to obtain density estimation for each cluster [7]. Similarly, mixture models covers the data well as the dominant patterns are captured by the components of the mixture model. A mixture model with high likelihood results in component distributions with high peaks, which means that the data in clusters are dense [36].

Traditional clustering algorithms such as k-means utilize unsupervised learning to group samples that are 'near' each other according to predefined measure of similarity [31]. These methods are more suitable for continuous data which has well defined distance measures. Although several similarity measures are defined for binary data, their application in binary data is not straightforward. Furthermore, our major application area was cancer genetics and cancer is not a single disease but a heterogeneous collection of several diseases. Mixture models are well-known for their ability to model heterogeneity [47]. In the current application we have used unsupervised clustering on cancer data sets with multiple cancer types, hence, one cluster can contain cancer types from multiple cancers. Mixture models also provide the facility of soft clustering, however soft clustering is out of the scope of this work.

3.2.2 Model selection in mixture models

Expectation Maximization (EM) algorithm can be used to learn the maximum likelihood parameters of the mixture model if the number of component distributions are known in advance [13]. However, the number of components (i.e. number of clusters) in the data is often unknown a priori in most real-world applications. Hence, model selection is also an essential prerequisite of learning mixture models. Model selection is the process of choosing a model of appropriate complexity that fits the given data set optimally [11, 23]. The complexity parameter in mixture model is the number of mixture components, therefore, model selection in mixture model is the choice of appropriate number of components in the mixture model.

A plethora of criteria have been proposed in the literature to determine the appropriate number of mixture model components [47]. For example, authors in [8], [17], and [56] comprehensively review deterministic, stochastic and resampling criteria to evaluate the performance of mixture model and therefore select the model of appropriate complexity. Deterministic criteria consists of Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Minimum Description Length (MDL), and integrated classification likelihood (ICL). Similarly, stochastic methods include Markov Chain Monte Carlo (MCMC), and resampling methods include bootstrapped likelihood ratio test [45], while authors in [80] propose a robust approach against model mis-specification leading to a better fitting mixture density based on minimum Hellinger distances. In addition, the authors in [10] and [29] use penalised likelihood method for model selection in mixture model.

A popular criterion measure of the quality of mixture models is the data likelihood [63]. In addition, cross-validation is widely used model validation technique. Therefore, we use cross-validated likelihood to select the model of appropriate complexity as documented in [67]. A mixture model with large number of mixture components produces larger value for the log–likelihood in Equation 2 for training data. However, a mixture model with large number of mixture components also overfits the data, and generalizes poorly on the future unseen data. Additionally, mixture models with large number of components require greater resources: both time and memory. In contrast, a mixture model with smaller number of mixture components results in an underfitted model, and is unable to adequately represent the underlying true data distribution. Therefore, model selection aims to optimize this trade-off between too simple and too complex models [47]. A well trained mixture model with appropriate number of mixture components estimates the underlying data distribution better and produces high likelihood values for the unseen data which is the primary objective of our model selection procedure [7].

3.3 Semantic pattern mining

The expansion of the semantic web and increasing availability of domain knowledge in the form of ontologies has resulted in the growth of semantic data. Consequently, ontologies are recognized as useful for encoding semantics of data also in the machine learning and data mining communities and recent studies have shown that additional knowledge can enhance the knowledge discovery process [57]. Note that—in contrast to the philosophical definition of *ontology*—we use the plural form *ontologies* to emphasize that they can be independent domain models, possibly obtained from different sources.

In our application area, ontologies come from known biological landmarks or other known biological information. Similarly, many application areas have readily available background information that could prove useful in the data analysis process, especially in biological and clinical applications. Semantic data mining addresses this challenge of mining the abundance of available knowledge encoded in domain ontologies to improve the process of data mining [74].

Existing semantic subgroup discovery algorithms are either specialized for a specific domain [69] or adapted from systems that do not take into the account the hierarchical structure of background knowledge [72]. On the other hand, the recently developed semantic subgroup discovery system Hedwig [73], is designed as a general purpose semantic subgroup discovery system that uses domain ontologies to structure the search space to formulate the hypotheses using ontology concepts.

Semantic subgroup discovery, as addressed by the Hedwig system, results in relational descriptive rules. Hedwig uses ontologies as background knowledge and training examples in the form of Resource Description Framework (RDF) triples. Formally, we define the semantic data mining task addressed in this work as follows.

Given:
- the empirical data in the form of a set of training examples expressed as RDF triples,
- domain knowledge in the form of ontologies, and
- an object-to-ontology mapping which associates each object from the RDF triplets 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.

Subgroup describing rules are first-order logical expressions. Consider the following rule, used to explain the format of induced subgroup describing rules, such as, for example: `Class(X) ← C1(X), R(X,Y), C2(Y)` with True Positives (TP)=80 and False Positives (FP)=20. Variables `X`, `Y` represent sets of input instances, R is a binary relation between the examples and C_1, C_2 are ontological concepts. This rule is interpreted as follows. If an example X is annotated with concept C_1, and is related with an example Y via R,

```
Input  : Input examples E, background knowledge B, target class value c, beam
         size k, p-value threshold α
Output: Set of rules
rules ← [default_rule(E, c, B)]
while improvement(rules) do
    // Add specializations of each rule to the beam
    for rule ∈ rules do
        extend(rules, specialize(rule, B))
    end
    rules ← best(rules, k) // Select the top k rules
end
rules ← validate(rules, α) // Significance testing
return rules
```

Algorithm 1: Hedwig's induce(E, B, c, k, α) procedure.

```
Input  : Rule to specialize rule, background knowledge B
Output: Set of specializations of rule
specializations ← []
// Predicates that can be specialized
eligible_preds ← eligible(predicates(rule))
for predicate ∈ eligible_preds do
    // Specialize by traversing the subClassOf hierarchy
    for subclass ∈ subclasses(predicate, B) do
        new_rule ← swap(rule, predicate, subclass)
        if can_specialize(new_rule) then
            append(specializations, new_rule)
        end
    end
    // Specialize by negating
    new_rule ← negate(rule, predicate)
    if can_specialize(new_rule) then
        append(specializations, new_rule)
    end
end
if rule ≠ default_rule then
    // Specialize by adding a new unary predicate
    new_predicate ← next_non_ancestor(eligible_preds)
    new_rule ← append(rule, new_predicate)
    if can_specialize(new_rule) and non_redundant(new_rule) then
        append(specializations, new_rule)
    end
end
if is_unary(last(predicates(rule))) then
    // Specialize by adding new binary predicates
    extend(specializations, specialize_binary(new_rule))
end
return specializations
```

Algorithm 2: Hedwig's specialize($rule$, B) procedure.

and Y is annotated with concept C_2, then the conclusion $Class(X)$ holds. This rule condition is true for 100 input instances ($TP + FP$, also called *coverage*), 80 of which are of the target class (TP, also called *support*).

The Hedwig system, which implements Algorithms 1 and 2 to search for interesting subgroups, supports ontologies and examples to be loaded as a collection of RDF triples (a graph). The system automatically parses the RDF graph for the `subClassOf` hierarchy, as well as any other user-defined binary relations. Hedwig also defines a namespace of classes and relations for specifying the training examples to which the input must adhere.

The algorithm uses beam search, where the beam contains the best N rules found so far. The search starts with the default rule which covers all the input examples. In every iteration of the search, each rule from the beam is specialized via one of the four operations:

1. Replace predicate of a rule with a predicate that is a sub-class of the previous one,
2. Negate predicate of a rule,
3. Append a new unary predicate to the rule,
4. Append a new binary predicate, thus introducing a new existentially quantified variable (note that the new variable needs to be 'consumed' by a literal to be conjunctively added to this clause in the next step of rule refinement).

Rule induction via specializations is a well-established way of inducing rules, since every specialization either maintains or reduces the current number of covered examples. A rule will not be specialized once its coverage is zero or falls below some predetermined threshold. When adding a new conjunction, we check that if the extended rule does not improve the probability of the conclusion (we use the redundancy coefficient, as in [30]), then it is not added to the pool of specializations. After the specialization step is applied to each rule in the beam, we select new set of the best scoring N rules. If no improvement is made to the collection of rules, the search is stopped. In principle, our procedure supports any rule scoring function. Numerous rule scoring functions (for discrete targets) are available: χ^2, precision, *WRAcc* [40], leverage and lift. The latter is the default choice and was also used in our experiments. After the induction phase, the significance of the findings is tested using the Fisher's exact test [18]. To cope with the multiple-hypothesis testing problem, we use Holm-Bonferroni [27] direct adjustment method with $\alpha = 0.05$.

3.4 Visualization using banded matrices

Consider a binary matrix M with N rows and d columns and two permutations, κ and π of the first N and d integers. Matrix M_κ^π, defined as $(M_\kappa^\pi)_{i,j} = M_{\kappa(i),\pi(j)}$, is constructed by applying the permutations π and κ on the rows and columns of M. If, for some pair of permutations π and κ, matrix M_κ^π has the following property:

– Each row i of the matrix has the *consecutive ones property*. This means that the column indices for which the value in the matrix is 1 appear consecutively, i.e. on indices $a_i, a_i + 1, \ldots, b_i$,
– For each i, we have $a_i \leq a_{i+1}$ and $b_i \leq b_{i+1}$,

then the matrix M is *fully banded*. Furthermore, if matrix M is fully banded, then its transpose $M^\top$ is also fully banded.

Figure 2 demonstrates the motivation behind banded matrices as it shows that finding the banded structure of a matrix simultaneously exposes the clustered structure of the underlying data. This means that banded matrix factorization can provide an evaluation of the clustering results – we expect that clusters, discovered in a data set, will also be exposed by the banded matrix visualization. Similar visual perspective can also be shown by displaying all the clusters together, however using independent banded matrices on them gives more validity to the results. Allowing the samples from the same cluster to spread along the matrix will ease pattern comparison as similar patterns from different clusters will be grouped together. Additionally, it is easier to see the similar clusters in the data and make future decisions such as splitting of clusters or merging of clusters for future experiments. When the reordering selected does not depend on the cluster structure discovered, the resulting figures offer new insight into both the data and the clustering.

For a fully banded matrix, it can be shown that a banded structure can be found in polynomial time [20]. We cannot expect, however, that real world matrices, especially those originating in a disease as heterogeneous as cancer, will be fully banded. The problem is that, for a matrix involving noise, finding the correct row and column permutations that show a structure, close to a banded one, may be computationally unfeasible. We therefore need algorithms that attempt not only to find column and row permutations that are as close to banded as possible in some sense, but also find these 'almost banded' structures in a decent time frame.

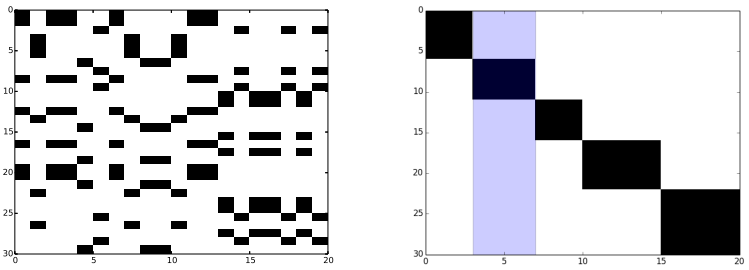


Fig. 2: An example of a binary matrix before and after row and column permutations exposing a banded structure.

The method used to find the banded structure of a matrix in this article, called the bidirectional minimum matrix augmentation (biMBA) method, was first proposed in [65] and was first used as a method of banded matrix extraction in [20]. One step of the method consists of three substeps,

1. Ensuring the maximum ones property. In the first step, each row of the matrix is transformed to have the consecutive ones property by finding the smallest number of matrix elements that have to be changed (either from 1 to 0 or from 0 to 1) in order for the row to have the consecutive ones property.

Theorem 1 *Given a matrix M, finding the correct elements to change in the i-th row of M is equivalent to solving the* maximum subarray problem *for the matrix W, defined as*

$$W_{i,j} = \begin{cases} +1 & \text{if } M_{i,j} = 1 \\ -1 & \text{if } M_{i,j} = 0 \end{cases}$$

Proof: The transformation of the matrix row i into one with a consecutive ones property is obviously an operation that results in the row having elements $a, a+1, \ldots, b$ set to 1 and the remaining elements set to 0 (for some pair of integers $1 \leq a \leq b \leq n$), so the task of finding this transformation is equivalent to finding the correct (those that require the smallest number of matrix element changes) values for a and b. The number of matrix element changes assigned to each value of (a, b) is equal to

$$C_i(a,b) = |\{j | a \leq j \leq b \wedge M_{i,j} = 0\}| + |\{j | (j < a \vee j > b) \wedge M_{i,j} = 1\}| \quad (3)$$

The task of finding the smallest number of matrix element changes to make the row have the consecutive ones property is therefore equivalent to finding $\text{argmin}_{a \leq b} C_i(a,b)$

On the other hand, solving the maximum subarray problem for the i-th row of matrix W is defined as finding the subarray of the matrix for which the sum of the elements is the biggest. Just as before, each subarray can be represented by two integers a, b which represent the start and end point of the subarray. The maximum subarray problem is equivalent to finding $\text{argmax}_{a \leq b} P_i(a,b)$, where P_i is defined as $P_i(a,b) = \sum_{i=a}^{b} W_{i,j}$. We know that the elements of W can only equal 1 or -1, so $P_i(a,b)$ can be rewritten as

$$|\{j | a \leq j \leq b \wedge W_{i,j} = 1\}| - |\{j | a \leq j \leq b \wedge W_{i,j} = -1\}| \quad (4)$$

which can, following the definition of $W_{i,j}$, be written as

$$P_i(a,b) = |\{j | a \leq j \leq b \wedge M_{i,j} = 1\}| - |\{j | a \leq j \leq b \wedge M_{i,j} = 0\}| \quad (5)$$

We now consider the fact that the set $S_i = \{j | M_{i,j} = 1\}$ (which is fixed for a given i) is the disjoint union of the sets $S_{i,\in} = \{a \leq j \leq b | M_{i,j} = 1\}$ and $S_{i,\notin} = \{(j < a \vee j > b) \wedge M_{i,j} = 1\}$ and we see (since $|S_i| = |S_{i,\in}| + |S_{i,\notin}|$) that

$$P_i(a,b) = |S_{i,\in}| - |\{j | a \leq j \leq b \wedge M_{i,j} = 0\}| \quad (6)$$

$$= |S_i| - |S_{i,\notin}| - |\{j | a \leq j \leq b \wedge M_{i,j} = 0\}| = |S_i| - C_i(a,b). \quad (7)$$

This shows that $P_i(a,b) = \text{const.} - C_i(a,b)$, meaning that $\text{argmin}(C_i) = \text{argmax}(P_i)$, concluding the proof.

Theorem 1 shows that for each row i, the elements which have to be changed to transform it into a consecutive ones row can be found by solving a *maximum subarray problem* which is solvable in linear time by finding, for each index j, the best subarray s_j ending at j. If $W_{i,j} = -1$, then s_j is obviously equal s_{j-1} with the addition of j (if the sum of the elements of s_{j-1} is positive) or it is an empty array with sum 0 (if the sum of the elements of s_{j-1} is zero). On the other hand, if $W_{i,j} = 1$, then adding j to the subarray s_{j-1} clearly makes the best possible subarray ending at j.

After transforming M into a matrix with the consecutive ones property, we denote the new matrix M'.

2. Ensuring the existence of a banded structure.

To ensure the existence of a banded structure for M', we now must further ensure that there is no pair of rows i_1, i_2 such that $a_{i_1} < a_{i_2}$ and $b_{i_2} < b_{i_1}$ (the interval of ones in row i_2 is *completely subsumed* by the interval of ones in row i_1). It is obvious that if such a pair exists, then M' is not fully banded, since according to a_{i_1}, a_{i_2}, the row i_1 should be above row i_2 ,but according to b_{i_1}, b_{i_2}, the row i_2 should be above row i_1. However, as shown in [20], the reverse also holds: if no such pair i_1, i_2 exists, then the matrix is fully banded.

This can be seen since if no such pair exists, we can sort the matrix rows by a_i, then by b_i to obtain a fully banded matrix M''. For any row i of M'' (with consecutive ones between a_i and b_i), we then know that if $a_i = a_{i+1}$, we will have $b_i \leq b_{i+1}$ by our sorting, and if $a_i < a_{i+1}$, then $b_i > b_{i+1}$ would mean that before sorting, row $i+1$ had an interval of ones that was completely subsumed by the interval of ones in row i, which is not possible.

In order to eliminate fully subsumed pairs of rows, in the second step, the algorithm finds each pair of rows i_1, i_2 such that $a_{i_1} < a_{i_2}$ and $b_{i_1} > b_{i_2}$. Then, for each such pair, the algorithm performs the minimum number of matrix element changes required so that either $a_{i_1} = a_{i_2}$ (this is done by adding ones *before* a_{i_2} to row i_2) or $b_1 = b_2$ (by adding ones *after* b_{i_2} to row i_2) or by completely deleting all ones in row i_2. Because all changes are made to row i_2, if we traverse the pairs i_1, i_2 in a double for loop, we can be sure that no completely subsumed intervals will be created anew, meaning that the result of this step is a fully banded matrix.

3. Finding the permutation to show the banded structure of M''. As we have shown in the previous two points, the matrix M'' is fully banded. Furthermore, there exists a permutation π of the rows of M'' that exposes the banded structure of M''. This permutation can be found by simply sorting the starting points of the intervals of ones in the rows of M'' from smallest to largest, resolving ties by the endpoints of the intervals (sorting first by a_i, then by b_i).

Following the steps outlined above, Algorithm 3 calculates the best possible (in some way) permutations of rows that will best expose the banded structure of the input matrix. The result of the method is the original matrix M, on which we apply the permutation π. However, the biMBA algorithm is

non-optimal, heuristic, and does not find any permutation of columns [20]. To find both a permutation of columns and rows, the *alternating* biMBA method transposes the resulting matrix and iteratively repeats the described method on the transposed matrix until either convergence or reaching a predetermined number of steps. The alternating biMBA method clearly finds both a permutation of rows and a permutation of columns, however it is still (like the biMBA method) non-optimal and heuristic in nature. Also, this second iterative step comes with some price for some of the data described in this article: in the first data set, where neighboring columns of a matrix represent chromosome bands that are in physical proximity to one another, the goal may be to only find the optimal row permutation while not permuting the matrix columns.

As motivated by Figure 2, finding a banded structure of a matrix will expose the cluster structure of the underlying data. The image of the banded structure can then be overlaid with a visualization of clusters, as described in Section 3.2. Because the rows of the matrix represent instances, highlighting one set of instances (one cluster) means highlighting several matrix rows. If the discovered clusters are exposed by the matrix structure, we can expect that several adjacent matrix rows will be highlighted, forming a wide band. Highlighting of clusters need not be limited to only one cluster: because each instance belongs to exactly one cluster, we can highlight them all at once. The only limitation is the number of clusters: because each cluster is colored with its own color, too many clusters may mean that colors will be too similar to each other to be distinguishable by the human eye.

The image of the clusters can also be overlaid with a visualization of the patterns explaining the clusters, presented in Section 3.3. If a chromosome band is discovered as an important chromosome band for the characterization of a cluster, we highlight the corresponding column. In the case of composite rules of the type Rule 1: `Cluster3(X)` $\leftarrow$ `1q43-44(X)` $\wedge$ `1q12(X)`,

Input : Input binary $n \times m$ matrix M
Output: Permutation π of rows of M such that M_π is approximatelly banded

```
// 1. Ensuring the maximum ones property
for i = 1, 2, ..., n do
    a_i, b_i = to_consecutive_ones(M_i) // After this step, the ones in row
        M_i appear in columns a_i, a_i + 1, ..., b_i
end
// 2. Ensuring the existence of a banded structure
for i = 1, 2, ..., n do
    for j = i, i + 1, ..., n do
        if a_j < a_i ∧ b_j > b_i then
            a_i, b_i =extend_or_delete(i, j)
        end
    end
end
// 3. Finding the permutation to show the banded structure of M''
π = argsort([(a_1, b_1), ..., (a_n, b_n)]) return π
```

Algorithm 3: The bidirectional MBA algorithm.

both bands are understood as equally important and are therefore both highlighted. If a chromosome band appears in more than one rule, this is visualized by a stronger highlight of the corresponding matrix column. In the case of the ideal example, shown in Figure 2, the second cluster is completely defined by having ones in columns $3, 4, 5, 6$, and 7. We show this by highlighting these columns in the banded matrix. It is to be noted that the banded matrix visualization helps to determine if the clustering results are plausible. It also helps to identify the similarities and differences between clusters with respect to the patterns in the data.

4 Experimental data

In this section, we present the data sets which were used in the experiments. We first present a detailed explanation of multiresolution chromosomal amplification data, followed by the presentation of selected publicly available data sets that were previously used in [61].

4.1 Multiresolution chromosomal amplification data

A wide range of genetic mutations and molecular mechanisms known as chromosomal aberrations have been identified as the hallmarks of various disorders such as cancer, schizophrenia, and infertility [5, 75]. In cancer research, identification and characterization of chromosomal aberrations are crucial to study and understand pathogenesis of cancer. Furthermore, study of chromosomal aberrations provides necessary information to select the optimal target for cancer therapy on an individual level [33]. Study of chromosomal aberrations also has several clinical applications such as studying multiple congenital abnormalities and identifying the family history of Down syndrome [54].

The data set we examined consists of DNA copy number amplifications in 4,590 cancer patients. The data describes 4,590 patients as data instances, with attributes being chromosomal locations indicating amplifications in the genome. These aberrations are described as 1's (amplification) and 0's (no amplification). Authors in [51] describe the amplification data set in detail. Amplification data is further described at two different resolution levels (312 and 393 locations, for 24 different chromosomes).

Given the complexity of the multiresolution data, we were forced to reduce the complexity of the learning setting to a simpler one, allowing us to develop and test the proposed methodology. To this end, we have reduced the size of the data set: from the initial set of instances describing 4,590 patients, each belonging to one of the 73 different cancer types, we have focused on 34 most frequent cancer types only, as there were small numbers of instances available for many of the rare cancer types. This reduced the data set from 4,590 instances to a 4,104 instances. The choice of 34 most frequent cancers is motivated by the fact that it covers 90% of the entire data set. Since the

original data with 393 genomic locations are high dimensional and the results could be greatly affected by the curse of dimensionality [6], we partitioned the data into 24 different chromosomes and process each chromosome at a time. Additionally, chromosome-wise processing may help us find chromosome specific patterns for different cancer types. Nevertheless, this division is based on the assumption that the effects of amplifications on different chromosomes, produced by a cancer type, are independent. Similar to the experiments in [25], which showed differences in frequent itemsets computed from one cluster at a time to the whole data set at once, we can expect different patterns when they are computer from one chromosome at a time to the whole data set at once.

In addition, in the experiments we have focused on a single chromosome (chromosome 1), using as input to step 2 of the proposed methodology the data clusters obtained at coarse resolution using a mixture modeling approach [52].

When chromosomes are extracted from the data, some cancer patients show no amplifications in any regions of the chromosome 1. We have removed such samples without amplifications (zero vectors) because we are interested in the amplifications and their relation to cancers, not their absence. Considering negation cases is unsuitable because we are only investigating one chromosome at a time. A negation result could infer that if a region is not aberrated, it is likely to be a specific cancer which will be misleading as information from other chromosomes are missing. These reduces the sample size, for example sample size of chromosome 1 is reduced from 4,104 to 407. While this data reduction may be an over-simplification, finding relevant patterns in this data set is a huge challenge, given the fact that even individual cancer types are known to consist of cancer sub-types which have not yet been explained in the medical literature. If we consider the entire data, inferencing and density estimation will produce degenerate results because of the curse of dimensionality [6]. Additionally, the experiments performed on chromosome 1 can be seamlessly extended to all the other chromosomes, thus efficiently using each and every sample present in the data. Furthermore, chromosomewise analysis can generate chromosome specific patterns for certain cancer types. The proposed methodology may prove, in future work, to become a cornerstone in developing means through which such sub-types could be discovered, using automated pattern construction and innovative pattern visualization using banded matrices visualization.

In addition to the DNA amplifications data sets, we used supplementary background knowledge in the form of an ontology to enhance the analysis of the data set. The supplementary background knowledge consists of hierarchical structure of multiresolution amplification data, chromosomal locations of fragile sites, virus integration sites, cancer genes, and amplification hotspots. The hierarchical structure of multiresolution data is due to International System of Cytogenetic Nomenclature (ISCN) which allows the exact description of all numeric and structural amplifications in genomes [62]. A fragile site is a chromosomal region that tends to show a constriction or a gap and may tend to break on metaphase chromosomes when subjected to partial replication stress, i.e. following partial inhibition of DNA synthesis [16]. A metaphase chromo-

some is a chromosome in the stage of the cell cycle (the sequence of events in the life of a cell) when a chromosome is most condensed, highly coiled, and aligned in the equator of the cell before being separated into each of the two daughter cells. At this stage chromosome is easiest to distinguish and study. Virus integration sites are also the chromosomal locations where viral DNA inserts into host-cell DNA [24]. Approximately, 12% of cancers are caused by viruses [24]. Cancer genes are also the chromosome locations of known cancer causing genes. The list was obtained from [19]. Amplification hotspots are frequently amplified chromosomal loci identified using computational modeling [51].

4.2 Publicly available data sets

In addition to the chromosomal amplifications data, we tested our methodology on four publicly available data sets originally used in [61].

- **Cities** data set describes the most and least liveable cities in the world according the Mercer ranking.
- **NY Daily** data set describes the crawled news items along with their sentiment scores.
- **Tweets** data set is a collection of tweets with different features where the original task is to identify different sports related tweets.
- **Stumble Upon** data set consists of training data set used in the Kaggle competition.

To generate the hierarchical features, the same 'DBpedia Direct Types' ontology was used in the first three experiments, and the 'Open directory project' ontology was used to extract categories for each URL in the fourth data set, i.e. we used the same approach as in the original experiments reported in [61].

Since the data sets were highly sparse, we preprocessed the data to remove highly sparse variables. In the Cities data sets, we selected only those features which were positive in at least 25 different samples, but also eliminated features that were very dense, i.e. those that were positive in more than 170 instances. In the NY Daily data sets, we selected only the features that were positive in more than 200 samples but less than 450 samples. In the Tweets data set we selected only the features that were positive in more than 100 samples of the Tweets data set. Finally, in the Stumble Upon data set we selected only the features that were positive in more than 400 samples. Such preprocessing was motivated by the fact that features that are either very sparse or too dense carry very little information for class discrimination. Moreover, by removing these features we also mitigate the negative curse of dimensionality effects [6].

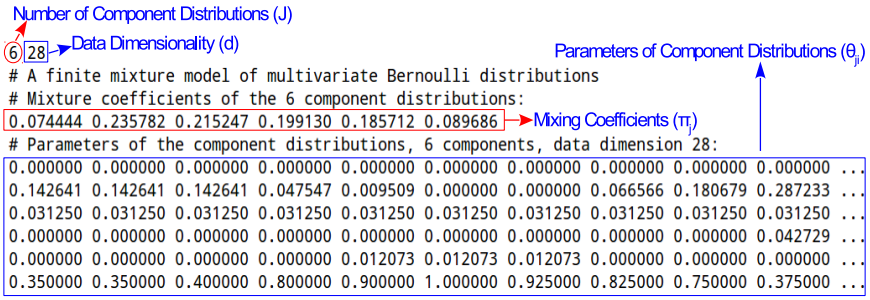


Fig. 3: Mixture model for chromosome 1. Only first 10 dimensions are shown for clarity. The figure just depicts a collection of numbers in the mixture model, which does not provide much insight to the expert.

5 Experiments on multiresolution chromosomal amplification data

The following sections describe the results of running the developed three-step methodology on the chromosomal amplification data. We present the experimental results, result visualization and interpretation.

5.1 Mixture modeling

Mixture modeling itself consists of three steps: first we need to use model selection to determine the number of components (i.e. clusters) in the mixture model. Second, we need to learn the parameters of each component distributions, and finally, use the selected model to generate the data clusters. For the chromosomal amplification data set, we used the mixture models trained in our earlier contribution [52]. Through a model selection procedure documented in [67], the number of components for modeling chromosome 1 was set to $J = 6$.

Figure 3 shows a visual illustration of the mixture model parameters for chromosome 1. In the figure, the first line denotes the number of components (J) in the mixture model and the data dimensionality (d). The lines beginning with # are comments and can be ignored. The fourth line shows the parameters of component distributions (π_j) which are six probability values summing to 1. Similarly, the last six lines of the figure denote the parameters of the component distributions (θ_{ji}). Figure 3 does not provide any insight into the data as it consists of many numbers and probability values. Therefore, we use banded matrix for visualization to demonstrate and evaluate the results produced by the mixture models and provide additional insights into the data set.

We clustered the data using the mixture model depicted in Figure 3. Whereas the mixture model defines a probability model for the generation

of data and can thus be used in soft clustering, allocating data vectors to the component densities that maximize the probability of data defines a hard clustering. Here, we focus on hard clustering of the samples of chromosomal amplification data, dividing the data set into six different clusters. The number of samples in each cluster are the following: |Cluster 1| = 30, |Cluster 2| = 96|, |Cluster 3| = 88, |Cluster 4| = 81, |Cluster 5| = 75, |Cluster 6| = 37.

5.2 Cluster visualization using banded matrices

We used the bidirectional minimal banded augmentation method, described in Section 3.4, to extract the banded structure in the data. As explained in Section 3.4, we decided to only allow permutations of rows of the data matrix. In Figure 4, the black color indicates ones in the data and white color denotes zeros in the data. The resulting figure is then overlaid with the 6 clusters, discovered in Section 5.1.

By exposing the banded structure of a matrix, Figure 4 allows a clear visualization of the clusters discovered in the data. Examination of Figures 3 and 4 show that each cluster captures amplifications in some specific regions of the genome. Both figures capture a phenomenon that the p-arm of chromosome 1 (left part of the figure) shows a comparatively smaller number of amplifications whereas the q-arm shows a higher number of amplifications.

In Figure 4, cluster 1 (component 1, π_1) is characterized by pronounced amplifications in the end of the q-arm (regions 1q32–q44) of chromosome 1. The figure also shows that samples in the second cluster (component 2, π_2) contain sporadic amplifications spread across both p and q-arms in different regions of chromosome 1. This cluster does not carry much information and contains cancer samples that do not show discriminating amplifications in chromosomes as the values of random variables are near 0.5. It is the only cluster that was split into many separate matrix regions. In contrast, cluster 3 (component 3, π_3) portrays marked amplifications in regions 1q11–44. Cluster 4 (component 4, π_3) shows amplifications in regions 1q21–25. Similarly, cluster 5 is denoted by amplifications in 1q21–25. The visualization with banded matrices in Figure 4 also draws a distinction between clusters number 4 and 5, which upon first viewing show no obvious difference to the human eye. Cluster 6 (component 6, π_6) is defined by pronounced amplifications in the p-arm of chromosome 1.

5.3 Rules induced through semantic pattern mining

Using the method described in Section 3.3, we induced subgroup descriptions for each cluster as the target class. For a selected cluster, all the other clusters represent the negative training examples, which resembles one-versus-all approach in multiclass classification. In this section, we discuss the results pertaining to clusters 1 and 3 (see Tables 1 and 2), while the rules for the other

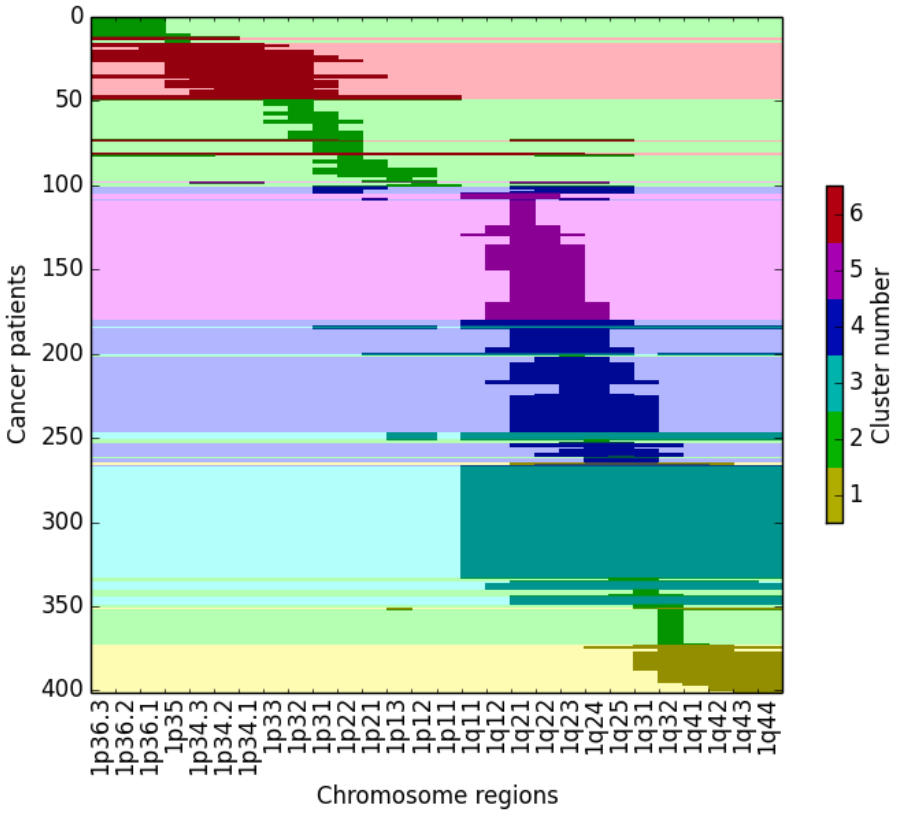


Fig. 4: Banded structure of the chromosome 1 data matrix with cluster information overlay.

#	Rules for cluster 1	TP	FP	Precision	Lift	p-value
1	`Cluster1(X)` ← 1q43–44(X)	26	88	0.23	3.09	0.000
2	`Cluster1(X)` ← 1q41(X)	26	90	0.22	3.04	0.000
3	`Cluster1(X)` ← 1q32(X)	24	116	0.17	2.33	0.000
4	`Cluster1(X)` ← HotspotSite(X)	30	280	0.10	1.31	0.000
5	`Cluster1(X)` ← FragileSite(X)	30	317	0.09	1.17	0.002

Table 1: Rules induced for cluster 1 of the chromosome 1 data set.

clusters, along with their visualization, are discussed in the following section. In our experiments we have considered only rules without negations in the rule conditions, as we are interested in the existence of amplifications characterizing the clusters and thereby the specific cancers (note that the absence of amplifications would namely characterize the absence of cancers not their presence).

#	Rules for cluster 3	TP	FP	Precision	Lift	p-value
1	Cluster3(X) ← 1q43--44(X) 1q12(X)	81	0	1.00	4.62	0.000
2	Cluster3(X) ← 1q11(X)	78	9	0.90	4.15	0.000
3	Cluster3(X) ← 1q43--44(X)	88	26	0.77	3.57	0.000
4	Cluster3(X) ← 1q41(X)	88	28	0.76	3.51	0.000
5	Cluster3(X) ← 1q12(X)	81	43	0.65	3.02	0.000
6	Cluster3(X) ← 1q32(X)	88	52	0.63	2.91	0.000
7	Cluster3(X) ← 1q31(X)	87	54	0.62	2.85	0.000
8	Cluster3(X) ← 1q25(X)	88	64	0.58	2.68	0.000
9	Cluster3(X) ← 1q24(X)	88	97	0.48	2.20	0.000
10	Cluster3(X) ← 1q21(X)	88	134	0.40	1.83	0.000
11	Cluster3(X) ← 1q22--24(X)	88	149	0.37	1.72	0.000
12	Cluster3(X) ← HotspotSite(X)	88	222	0.28	1.31	0.000
13	Cluster3(X) ← CancerSite(X)	88	245	0.26	1.22	0.000
14	Cluster3(X) ← FragileSite(X)	88	259	0.25	1.17	0.000

Table 2: Rules induced for cluster 3 of the chromosome 1 data set.

Tables 1 and 2 show the rules induced for clusters 1 and 3, together with their relevant statistics. The rules presented in Table 2 quantify the clustering results obtained in Section 5.1 and confirmed by banded matrix visualization in Section 5.2. The mixture model depicted in Figure 3 and banded matrix visualization depicted in Figure 4 show that cluster 3 is marked by the amplifications in the regions 1q11–44. However, the rules obtained in Table 2 show that amplifications in all the regions 1q11–44 do not equally discriminate cluster 3. For example, rule Rule 1: Cluster3(X) ← 1q43-44(X) ∧ 1q12(X) characterizes cluster 3 best with a precision of 1. This means that amplifications in regions 1q43–44 and 1q12 denote cluster 3. It also covers 81 of the 88 samples in cluster 3. Clinically, the amplifications in these regions characterises Ependymoma [52].

Nevertheless, amplifications in regions 1q11–44 shown in Figure 3 as discriminating regions, appear in at least one of the rules in Table 2 with varying degree of precision. The first part of the rule (i.e. amplifications in region 1q43–44) is the most discriminating for cluster 1 as shown in Table 1. However, with considerably reduced precision and lift.

Although the rule: Rule 2: Cluster1(X) ← 1q43-44(X) appears in semantic descriptions of both the clusters 1 and 3, addition of a conjunct 1q12 in the rule improves the discriminating power for cluster 3. Rule 2 covers all 88 samples of cluster 3 with precision of 0.77 whereas it covers 26 out of 30 samples in cluster 1 with the precision of 0.23. This shows that amplifications in region 1q43–44 characterize both clusters 1 and 3. If the negation rules are considered, amplifications only in regions 1q43-44 would more likely make it a candidate for cluster 1. Similarly, the second most discriminating rule for cluster 3 is: Rule 2: Cluster3(X) ← 1q11(X) which covers 78 positive samples and 9 negative samples.

The rules listed in Table 2 also capture the multiresolution phenomenon in the data. We input only one resolution of data to the algorithm but the

hierarchy of different resolutions is made available to the algorithm as background knowledge. For example, the literal 1q43–44 denotes a joint region in coarse resolution thus showing that the algorithm produces results at different resolutions. The results at different resolutions improve the understandability and interpretability of the rules [26].

Furthermore, other information added to the background knowledge are amplification hotspots, fragile sites, cancer genes, which are discriminating features of cancers but do not show to discriminate any specific clusters present in the data. Therefore, such additional information can be better utilized in situations where the data set contains not only cancer samples but also control samples which is unfortunately not the situation here as our data set has only cancer patients.

5.4 Visualizing semantic rules and clusters with banded matrices

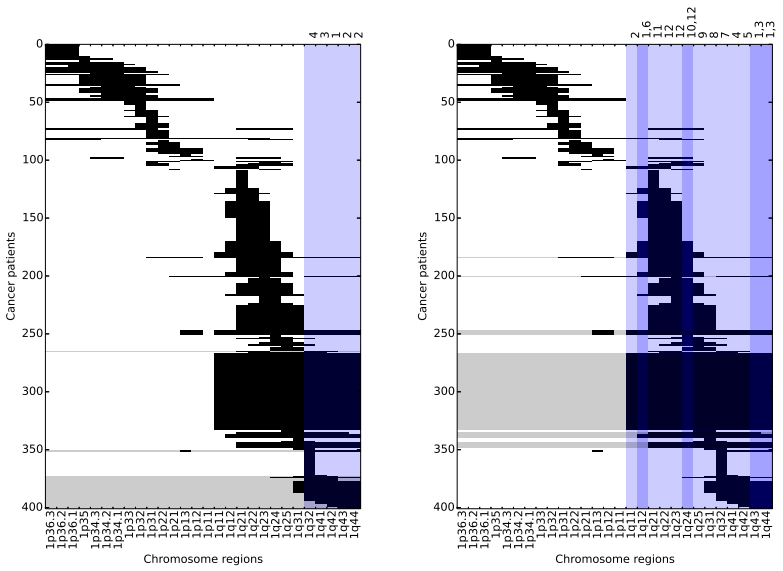


Fig. 5: Clusters 1 (left) and 3 (right) of the chromosome 1 data set with relevant columns highlighted. A highlighted column denotes that an amplification in the corresponding region characterizes the instances of the particular cluster. A darker hue means that the region appears in more rules. The numbers on top right of the figures correspond to rule numbers. For example, $1, 3$ above rightmost column of cluster 3 indicates that the chromosome region appears in rules 1 and 3 tabulated in Table 2.

The second way we can use the exposed banded structure of the data is to display columns that were found to be important due to appearing in rules from Section 5.3. We achieve this by highlighting the chromosomal regions which appear in the rules. Figure 5 depicts colored overlays of the rules on the ordered/serialized patient-chromosome matrix. As shown in Figure 5, the highlighted band for cluster 1 spans chromosome regions 1q32–44. For cluster 3, the entire q-arm of the chromosome is highlighted, as indeed the instances in cluster 3 have amplifications throughout the entire arm. We can see that the regions 1q11–12 and 1q43–44 appear in rules with higher lift, in contrast to the other regions. This tells us that the amplifications on the edges of the region are more important for the characterization of the cluster.

#	Rule	TP	FP	Precision	Lift	p-value
1	`Cluster2(X) ← 1p31(X)`	28	26	0.52	2.20	0.000
2	`Cluster2(X) ← 1p32(X)`	19	35	0.35	1.49	0.023

Table 3: Rules for cluster 2 of the chromosome 1 data set.

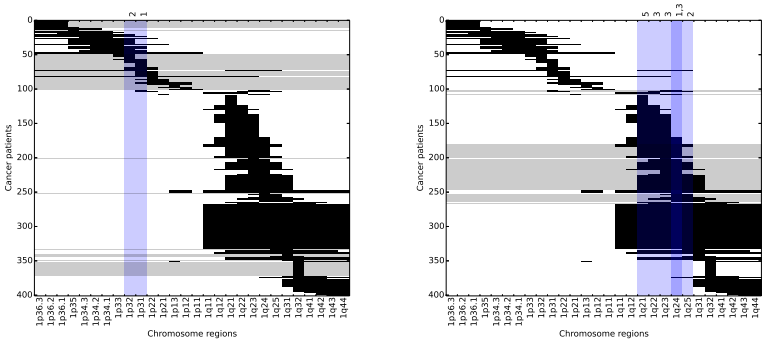


Fig. 6: Clusters 2 (left) and 4 (right) of the chromosome 1 data set with relevant columns highlighted.

As shown in the left panel of Figure 6, cluster 2 captures the heterogeneity in data. Since, we are using only chromosome 1, this cluster is more likely to capture those cancers that are characterized by amplifications in chromosomes other than chromosome 1. The samples from clusters are distributed in different parts by the banded matrix visualization. The amplifications captured by this cluster are miscellaneous samples, i.e. those cancers that do not show prominent amplifications in chromosome 1. Nevertheless, amplifications captured by this cluster characterize glioblastoma multiforme [52].

As shown in the right panel of Figure 6, cluster 4 captures the amplifications near the beginning of the q arm of chromosome 1. The rules tabulated in

#	Rule	TP	FP	Precision	Lift	p-value
1	`Cluster4(X) ← 1q24(X)`	81	104	0.44	2.20	0.000
2	`Cluster4(X) ← 1q25(X)`	57	95	0.38	1.88	0.000
3	`Cluster4(X) ← 1q22-24(X)`	81	156	0.34	1.72	0.000
4	`Cluster4(X) ← HotspotSite(X)`	81	229	0.26	1.31	0.000
5	`Cluster4(X) ← 1q21(X)`	56	166	0.25	1.27	0.000
6	`Cluster4(X) ← CancerSite(X)`	81	252	0.24	1.22	0.000
7	`Cluster4(X) ← FragileSite(X)`	71	276	0.20	1.03	0.001

Table 4: Rules for cluster 4 of the chromosome 1 data set.

Table 4 show amplifications in regions 1q21–1q25. Clinically, the amplifications in these regions of cluster 4 mark liposarcoma [52].

#	Rule	TP	FP	Precision	Lift	p-value
1	`Cluster5(X) ← 1q21(X)`	75	147	0.34	1.83	0.000
2	`Cluster5(X) ← 1q12(X)`	33	91	0.27	1.44	0.002
3	`Cluster5(X) ← 1q22-24(X)`	60	177	0.25	1.37	0.000
4	`Cluster5(X) ← HotspotSite(X)`	75	235	0.24	1.31	0.000
5	`Cluster5(X) ← CancerSite(X)`	75	258	0.23	1.22	0.000
6	`Cluster5(X) ← FragileSite(X)`	75	272	0.22	1.17	0.000

Table 5: Rules for cluster 5 of the chromosome 1 data set.

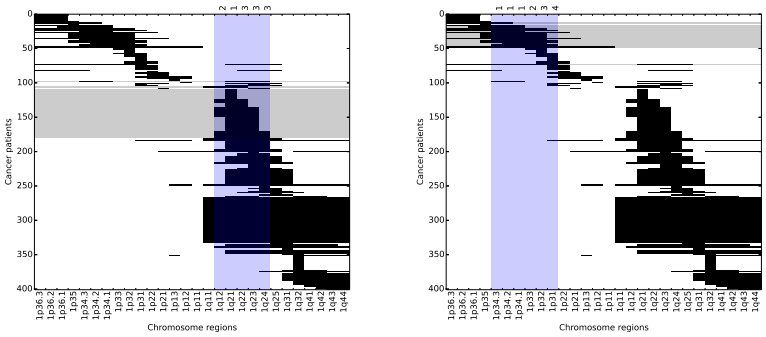


Fig. 7: Clusters 5 (left) and 6 (right) of the chromosome 1 data set. with relevant columns highlighted.

The regions and rules in Cluster 5, depicted in the left panel of Figure 7 overlap with the rules describing clusters 4. However, the rules describing these clusters have higher precision than those describing clusters 4. These two clusters are the prime candidates if any two clusters need to be merged.

In terms of clinical relevance, the amplifications the regions captured by this cluster denotes malignant fibrous histiocytoma of bone [52].

#	Rule	TP	FP	Precision	Lift	p-value
1	`Cluster6(X) ← 1p34(X)`	37	8	0.82	9.04	0.000
2	`Cluster6(X) ← 1p33(X)`	31	12	0.72	7.93	0.000
3	`Cluster6(X) ← 1p32(X)`	29	25	0.54	5.91	0.000
4	`Cluster6(X) ← 1p31(X)`	15	39	0.28	3.06	0.000
5	`Cluster6(X) ← CancerSite(X)`	36	297	0.11	1.19	0.000

Table 6: Rules for cluster 6 of the chromosome 1 data set.

The amplifications in the p-arm of Chromosome 1 captured by cluster 6 are depicted in the right panel of Figure 7. This is clearly distinguishable from other clusters because other clusters mainly capture the amplifications in q-arm of chromosome 1. The amplification in these regions characterizes the phenomenon of small cell lung cancer [52].

In summary, Figures 4 and 5 together offer view of the structure of the underlying data that is much more informative than simply the list of rules in Table 6 or the cluster visualization of Figure 3. The figures shows that most the samples in the same cluster also appear together in the banded matrix visualization even when we only allow permutations of rows in the data set. The figure, achieved by reordering the matrix rows by placing similar items closer together to form a banded structure, allows an easier visualization of the clusters and rules. It is important to reorder the rows independently of the clustering process. This is because the reordering does not depend on the cluster structure discovered. Therefore, the resulting figures offer new insight into both the data and the clustering.

6 Experiments on publicly available data sets

We repeated the experiments, using the developed pipeline on the publicly available data sets. In this section, we present the experimental results, their visualizations and interpretations for the four publicly available data sets.

6.1 Mixture modeling

Similar to the chromosome amplification data, we repeated the three steps (determining the number of clusters, learning the parameters of each component distribution and using the selected model to generate the clusters) for each of the publicly available data sets.

Following our previous work in [52], we used ten-fold cross-validation with cross-validated likelihood as the criteria for selection of the optimal number of clusters, similar to [67]. In each data set, we trained mixture models in a

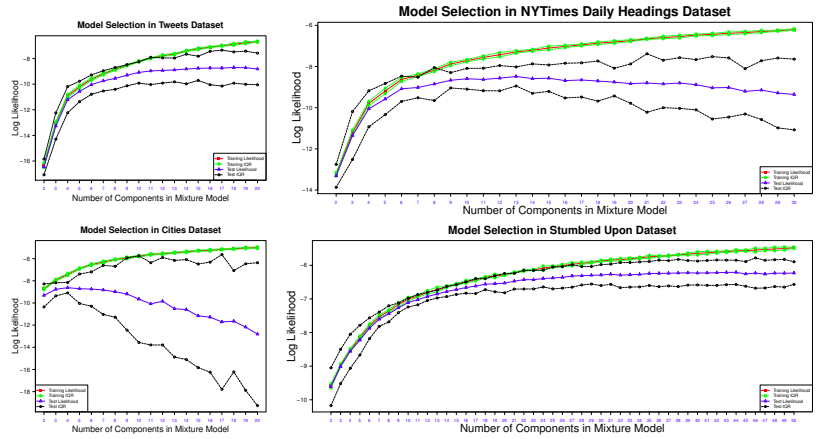


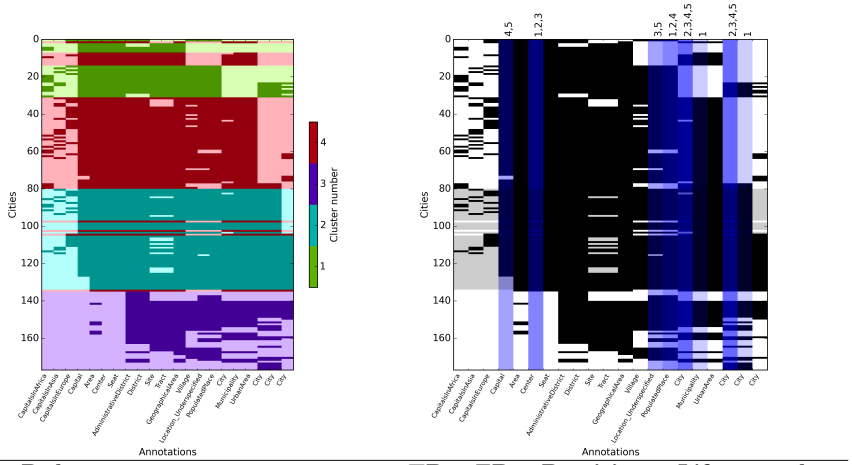
Fig. 8: Model selection using ten-fold cross-validation in NY Daily, Tweets, Stumble Upon, and Cities data sets. The figure depicts averaged log-likelihood for training and validation sets. The interquartile range (IQR) for 50 different training and validation runs in ten-fold cross-validation setting have also been plotted. The number of clusters is determined as the point at which training and validation likelihoods depict a peak.

cross-validation setting for the number of components ranging from 2 to 20 (30 and 50 in larger data sets NY Daily and Stumble Upon), with the assumption that there are at least two clusters in the data. Similarly, another assumption is that components greater than 20 (30 and 50 in NY Daily and Stumble Upon data) would overfit the data. Mixture models are susceptible to local optima, therefore, we train multiple models with the same number of components (50 in our experiments).

Figure 8 shows that for small numbers of clusters, the likelihood of mixture models increases smoothly until reaching a noticeable peak. For ideal data sets (seen in [67]), the peak represents a global maximum. Our experiments on real-world data sets show that identifying structures within data sets is not straight-forward. However, taking parsimony into account, even if larger numbers of components produce higher validataion likelihoods, we would select mixture models with a smaller number of components as they are computationally easier to train both in terms of time and memory and are also easily interpretable by the domain experts [26].

By determining the smallest number of components for which the likelihood as seen in Figure 8 of mixture models reaches a local peak, we select 6, 7, 4, and 10 components in the Tweets, NY Daily, Cities, and Stumble Upon data sets, respectively. Like in the case of chromosomal amplification data, we used the mixture model parameters for each data set to cluster it. We focused on hard clustering of the samples, dividing the data set into the number of clusters, determined in the previous step.

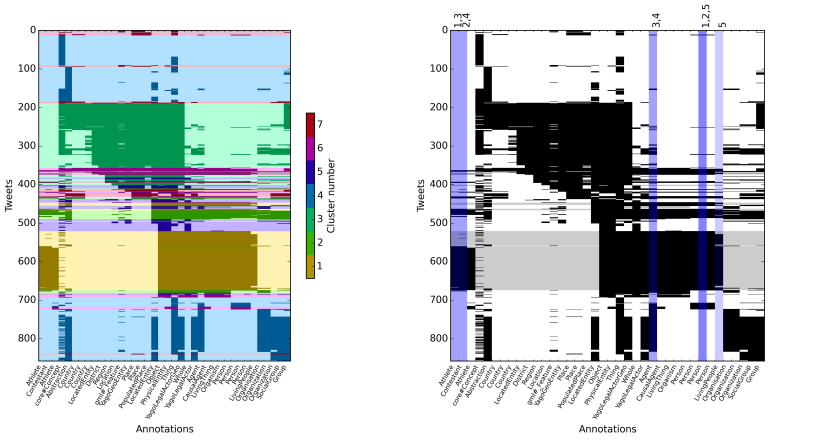
6.2 Cluster visualization using banded matrices



#	Rule	TP	FP	Precision	Lift	p-value
1	Cluster2(X) ← City(X), PopulatedPlace(X), Center(X), Municipality(X)	51	0	1.00	3.47	0.00
2	Cluster2(X) ← PopulatedPlace(X), City(X), Place(X), Center(X)	49	0	1.00	3.47	0.00
3	Cluster2(X) ← Underspecified(X), City(X), Place(X), Center(X)	48	0	1.00	3.47	0.00
4	Cluster2(X) ← PopulatedPlace(X), City(X), Place(X), Capital(X)	42	0	1.00	3.47	0.00
5	Cluster2(X) ← Underspecified(X), City(X), Place(X), Capital(X)	41	0	1.00	3.47	0.00

Fig. 9: The results of the methodology for the Cities data set. Top left: the banded structure of the Cities data matrix with cluster information overlay. Top right: cluster 1 of the Cities data set with relevant columns highlighted. Bottom: Rules for cluster 1 of the Cities data set.

On the publicly available data sets, we ran the alternating biMBA method to expose the banded structure of the matrices. The choice of alternating method was motivated by the fact that the ordering of the columns in the publicly available data sets was arbitrary. This is unlike the amplification data set which had fixed ordering of regions in the genome.

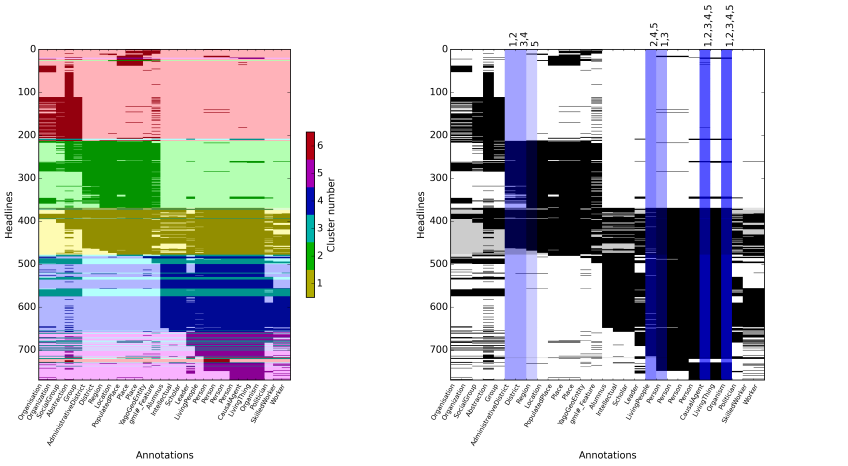


#	Rule	TP	FP	Precision	Lift	p-value
1	`Cluster1(X) ← Person(X) Athlete(X)`	117	30	0.80	4.21	0.00
2	`Cluster1(X) ← Person(X) Contestant(X)`	120	32	0.79	4.18	0.00
3	`Cluster1(X) ← Agent(X) Athlete(X)`	117	32	0.79	4.16	0.00
4	`Cluster1(X) ← Agent(X) Contestant(X)`	120	34	0.78	4.13	0.00
5	`Cluster1(X) ← Person(X) LivingPeople(X)`	145	46	0.76	4.02	0.00

Fig. 10: The results of the methodology for the Tweets data set. Top left: the banded structure of the Tweets data matrix with cluster information overlay. Top right: cluster 1 of the Tweets data set with relevant columns highlighted. Bottom: Rules for cluster 1 of the Tweets data set.

Cities. The biMBA algorithm converged after 7 iterations exposing the banded structure of the matrix. The banded structure in Figure 9 clearly visualizes the four clusters found by the presented methodology. Clusters 2 and 3 are almost completely separated from clusters 1 and 4. The visualization also shows that cluster 1 and cluster 2 are both composed of two parts which are hard to distinguish. This phenomenon was also captured during model selection in the Cities data set because the increase in validation likelihood was minimal when the number of components was increased from 3 to 4. When we selected four components, a relatively homogeneous cluster is broken down into two.

Tweets. The biMBA algorithm converged after 33 iterations for the Twitter data set with credible results. The visualization provided in Figure 10 shows that clusters 1, 2 and 3 are clearly separable from the rest of the data set. Cluster 4, the largest of the clusters, is split into two large parts, both of which are fairly homogeneous. However, clusters 5, 6, and 7 are relatively



#	Rule	TP	FP	Precision	Lift	p-value
1	`Cluster1(X) ← Agent(X) Admin.District(X) Organism(X)`	90	0	1.00	7.20	0.00
2	`Cluster1(X) ← Organism(X) Admin.District(X) LivingPeople(X)`	87	0	1.00	7.20	0.00
3	`Cluster1(X) ← Agent(X) District(X) Organism(X)`	92	1	0.99	7.12	0.00
4	`Cluster1(X) ← Organism(X) District(X) LivingPeople(X)`	89	1	0.99	7.12	0.00
5	`Cluster1(X) ← Organism(X) Region(X) LivingPeople(X)`	92	2	0.98	7.04	0.00

Fig. 11: The results of the methodology for the NY Daily data set. Top left: the banded structure of the NY Daily data matrix with cluster information overlay. Top right: cluster 1 of the NY Daily data set with relevant columns highlighted. Bottom: Rules for cluster 1 of the NY Daily data set.

small with the value mixture components equal to 0.07, 0.05, and 0.03. Hence, these clusters are not fully exposed in the visualization.

NY Daily. The biMBA algorithm converged after 11 iterations for the NY Daily data set. As seen in Figure 11, it clearly highlights clusters $1, 2$ and 6 and shows that clusters 4 and 3 are more similar to each other. Interestingly, even though cluster 3 is split into several parts, it can still be seen that the annotations, drawn on the left side of the visualization, are more important for cluster 3 (meaning that splitting the two clusters was a good choice). As in cluster 2 of the amplification data sets, the algorithm also highlights cluster 5 which does not capture a specific pattern but patterns scattered across different columns in the data set.

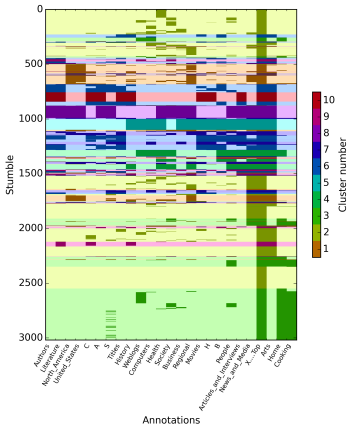


Fig. 12: The weakly banded structure of the Stumble Upon data set with cluster information overlay. Both the number of clusters and lack of a highly visible banded structure suggest a lack of structure in the data set.

Stumble Upon. The Stumble Upon data set was the only data set on which our methodology did not achieve credible results. The model selection procedure shows that both training and validation likelihood smoothly increases until the number of components is 20. Even after the number of components was greater than 20, even the validation likelihood did not decrease showing that there is no apparent structure in the data as depicted in the bottom right panel of Figure 8. The figure does not show variation in likelihood is not high among number of components and also within each component among different runs because the number of data samples are high to constrain the mixture model. Similarly, the biMBA algorithm converged much more slowly than in the other data sets, taking 521 iterations to reach the optimal banded structure. Visualizing the structure shows that the data is fractured into several small chunks. Some clusters, like 8 and 10, are separated from the rest, but the remaining clusters are sporadically scattered across all the rows.

6.3 Rules induced through semantic pattern mining

We ran the same semantic subgroup discovery procedure (with the same parameters) on the publicly available data sets as on the amplification data set. Due to the large amount of experimental results, we chose to describe one cluster and the top five rules for that cluster for each data set (Figures 9, 11 and 10). For the Stumble Upon data set, we did not describe the discovered cluster with rules because both the clustering and the banded structure visualization performed poorly on the data set.

Cities. This cluster was chosen as an example of a very well characterized cluster (Figure 9). We report the top five rules, all which have 100% precision. The first rule actually perfectly describes the cluster, since it covers all examples from cluster 2. By investigating the rule conjuncts it follows that this cluster contains cities that are at the same time annotated as centers, municipalities and populated places. Furthermore, the cities data set comes with a lable discribing its livability: low, medium, and high [61].

Although clustering, rule extraction, and visualization were performed independent of the labels, the rules and clusters mostly describe cities with medium and high livability. In the table we omit the full concept URIs for visual clarity. Nevertheless, the exact semantics of each concept can be verified by visiting the corresponding DBpedia pages, e.g., full URI of Center is `http://dbpedia.org/class/yago/Center108523483`.

NY Daily. For this data sets, we report the top five rules for cluster 1 (Figure 11). Similar to the previous data sets, the found rules are of high precision and each covers a relatively large portion of all examples from this cluster (a total of 107 examples). Compared to the subgroup descriptions found for the other five clusters, this cluster contains mainly headlines annotated with the District and Region concepts, together with Agent and Organism concepts.

Tweets. For this data set we feature the top five rules for cluster 1 (Figure 10). The rules found were of lower precision (76%-80%), which indicates that this cluster is harder to describe compared to the clusters mentioned in the previous two data sets. Nevertheless, the subgroup descriptions indicate that this cluster contains mainly tweets mentioning specific athletes (i.e. annotated with Person and Athlete concepts), and not for example teams or organizations, which do appear in rules for the other clusters (e.g., Organization concept). Furthermore, the tweets data set consists of associated class labels with denotes sports related and unrelated tweets [61]. Although, clustering, rule extraction, and visualization were performed independent of the label, this cluster mostly contains tweets related to sports.

6.4 Visualizing semantic rules and clusters with banded matrices

Similar to the chromosomal amplifications data sets, we also highlighted the relevant variables captured by the rules describing each cluster on the public data sets. We visualized the top 5 rules for the three publicly available data sets on which the rule discovery algorithm was run (the NY Daily, Cities and Tweets data sets).

Cities. Cluster 2 in the data set was perfectly described by the rules, This cluster was chosen as an example of a very well characterized cluster (Figure 9). The visualization shows a clear band of features, with the top instances annotated by features on the left side of the chart and the bottom instances annotated by features on the right. Cluster 2 in the middle is characterized by containing instances that are annotated by features on both sides of the band, as instances above it are not annotated by the rightmost features and instances above are not annotated by the leftmost features. The visualization shows that all five top rules cover features on both sides of the band.

NY Daily. For this data sets, we report the top five rules for cluster 1 as shown in Figure 11. The visualization clearly identifies the banded structure of the data, with three distinct vertical bands. The cluster is characterized as the cluster which contains instances, annotated by the features in the (unlike clusters 3 and 4) second and (unlike clusters 2 and 6) third band. The visualisation shows that all rules take this into account as all rules explain cluster 1 with at least one conjunct covering features on the second band and one conjunct in the third band.

Tweets. For this data set we feature the top five rules for cluster 1 (Figure 10). Despite the lower precision of rules, extracted by our methodology, the visualization still clearly shows the most important features for cluster 1. The banded structure visualization shows us two sets of features that are important to cluster 1. The first is the block of tweets, annotated with the annotations `Athlete` and `Contestant`. One of these two annotations features in all top four rules, found for this cluster. The second, larger block of features is used in all top five rules we present. Additionally, the visualization of all clusters can also tell us why the precision of rules, found for this data set, was lower: cluster 2 contains several instances which are annotated by all features that also annotate features in cluster 1.

7 Stability analysis of clustering results

The success of the presented three-part methodology depends upon the results of cluster analysis. Since mixture models and clustering are unsupervised, which might result in different clustering solutions in different runs of the algorithm [76]. Therefore, it is imperative that we evaluate the stability of the results produced by our mixture models. In our experiments, we use the Expectation Maximization (EM) algorithm to learn the maximum likelihood parameters of those mixture models. An important property of Expectation Maximization algorithm is that it is deterministic for a given data set and a given initialization [46]. In other words, given the same data sets and same initialization, EM algorithm always converges on the same final model. However, one of the drawbacks of Expectation Maximization algorithm is that it is susceptible to local optima [47]. Therefore, we use train the mixture model from random initialization multiple times to get the final result. In model selection, we consider the mean of the results and the dispersion to select the optimal number of components. In preparing the final model to use it for clustering, we train 200 different models from random initialization and select the one that produces the best likelihood as the final model for clustering.

We have the empirically evaluated the stability of our clustering results. We initially trained 100 mixture models initialized at random to convergence on the same data and measure the clustering accuracy, i.e. how often two observations belong to the same cluster. We could assume this setting to be a classification where first clustering solution to be the known class labels

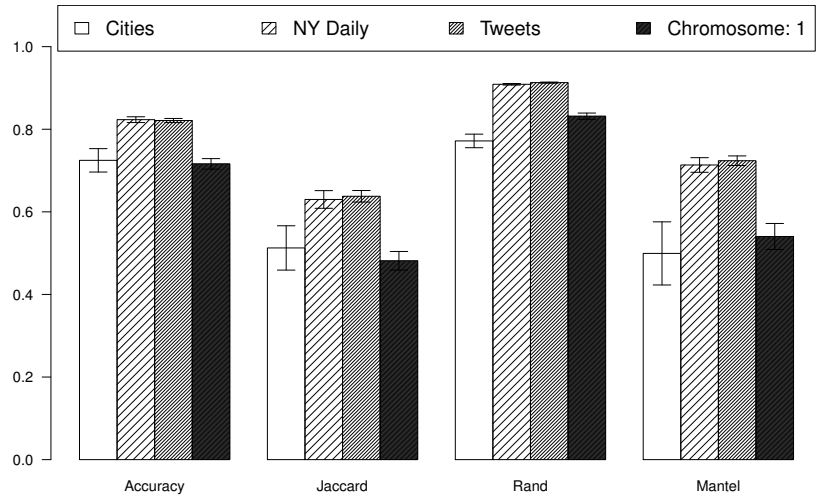


Fig. 13: Stability analysis of clustering results using four external measures of clustering.

and each subsequent clustering labels as the classification produced by the model [76]. Since, we compare 100 models to each other there will be 4950 comparisons in all. In addition to clustering accuracy, we have also calculated other external measures of clustering quality such as the Jaccard index, Rand index, and Mantel statistics to determine the similarity in different clustering results produced by differently trained mixture models.

Results in Figure 13 show that for unsupervised methods such as mixture models, the results of clustering are very stable. Clustering accuracy of approximately 70% is a very good result in a multiclass classification setting. For example the cities data set has 4 clusters, so a random classifier would generate only 25% accuracy. Jaccard index and Rand index of more than 50% also show that results are considerably stable. We calculate the Mantel statistics on the clustering results. The distance input to Mantel statistics is calculated from clustering labels obtained from two different clusterings. If two samples are in the same cluster, distance between them is $0, 1$ otherwise. The matrices are positively correlated, and the associated p-values are 0.001.

8 Summary and conclusions

The main contribution of this work is a three-part methodology for data analysis, consisting of (i) data clustering with mixture models, (ii) extraction of

semantic patterns (rules) from the clusters, using an ontology of relationships between the different resolutions of the multiresolution data, and (iii) integration of the results in a visual display, illustrating the clusters, and the identified rules by visualizing them over the banded matrix structure. The proposed visualization allows us to explain the discovered patterns by combining different views of the data, which may be difficult to compare without a unifying visual display. In our experiments, we analyzed DNA copy number amplifications in the form of 0-1 data, where the clustering developed in previous work was augmented by explanatory rules derived from a semantic pattern mining approach combined by the facility to display the bandedness structure of the data.

Our experiments with using the proposed algorithm on the NY Daily, Tweets and Cities data sets also demonstrate the wide usability of the algorithm which extends beyond the original application to DNA copy number amplifications onto any data set annotated by a hierarchically ordered set of background knowledge nodes. The results on the Stumble upon data set, while at first glance a negative result, also give important insight into the data set. Because all three algorithms (clustering, rule search and banded visualization) performed equally badly on the data set, we can with a much higher confidence claim that no particular structure in exists in the data set.

The proposed semi-automated methodology provides complete analysis of a complex real-world multiresolution data. The results produced in the form of different clusters, rules, and visualizations with the help of banded matrices are made interpretable for the domain experts. Especially, the visualizations with banded matrix helps to understand the clustering results and the rules generated by the semantic pattern mining algorithm. Furthermore, the background knowledge used to supplement semantic data mining algorithm enables us to analyze multiresolution data and garner results at different levels of multiresolution hierarchy. Similarly, the rules obtained by semantic data mining algorithm helps to quantitatively prioritize chromosomal regions that are hallmarks of certain cancers among different chromosomal regions that are amplified in those cancer patients.

The proposed approach accepts as input single-resolution data but allows for multiresolution data analysis due to the hierarchy of regions used as background knowledge in semantic pattern mining algorithm. In the future, we plan to develop a system to directly accept multiresolution data as input. Similarly, we will consider the cancer instance labels, since in the present work we focused only on cluster labels. In future work, we plan to formulate the problem as a multiclass classification problem in the semantic pattern mining setting as learning from ambiguous labels or partial labels and in mixture model clustering setting as soft clustering problem. Furthermore, another direction of research is to reformulate the banded matrix problem to consider class labels and directly benefit from cancer or cluster labels.

Similarly, we could also reformulate the instance descriptions by adding the truth values of the pattern alongside the original attributes and then compute the mixture model. Furthermore, the methodology is evaluated on data sets

(different data sets denoting different chromosomes) on a single application area, i.e. chromosomal amplifications in cancer genomics.

Acknowledgments

This work was supported by Helsinki Doctoral Programme in Computer Science — Advanced Computing and Intelligent Systems (Hecse) and by the Slovenian Ministry of Higher Education, Science and Technology grants. Additionally, the work was supported by the Academy of Finland (grant number 258568) and partially supported by the European Commission through the Human Brain Project (Grant number 604102). Thanks to Petar Ristoski for providing us with his data set exports.

References

1. P. R. Adhikari and J. Hollmén. Patterns from multiresolution 0-1 data. In *Proceedings of the ACM SIGKDD Workshop on Useful Patterns (UP '10)*, pages 8–16, New York, NY, USA, 2010. ACM.
2. P. R. Adhikari and J. Hollmén. Mixture models from multiresolution 0-1 data. In J. Fürnkranz, E. Hüllermeier, and T. Higuchi, editors, *Proceedings of Sixteenth International Conference on Discovery Science (DS 2013)*, volume 8140 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin Heidelberg, 2013.
3. P. R. Adhikari, A. Vavpetič, J. Kralj, N. Lavrač, and J. Hollmén. Explaining mixture models through semantic pattern mining and banded matrix visualization. In S. Džeroski, P. Panov, D. Kocev, and L. Todorovski, editors, *Discovery Science*, volume 8777 of *Lecture Notes in Computer Science*, pages 1–12. Springer International Publishing, 2014.
4. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, San Francisco, CA, USA, 12-15 September 1994. Morgan Kaufmann Publishers Inc.
5. D. G. Albertson. Gene amplification in cancer. *Trends in Genetics*, 22(8):447–455, 2006.
6. R. E. Bellman. *Adaptive control processes - A guided tour*. Princeton University Press, Princeton, New Jersey, U.S.A., 1961.
7. C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
8. G. Celeux. Mixture Models for Classification. In R. Decker and H-J. Lenz, editors, *Advances in Data Analysis*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 3–14. Springer Berlin Heidelberg, 2007.

9. C.-H. Chen, H.-G. Hwu, W.-J. Jang, C.-H. Kao, Y.-J. Tien, S. L. Tzeng, and H.-M. Wu. Matrix visualization and information mining. In J. Antoch, editor, *Proceedings in Computational Statistics (COMPSTAT 2004)*, pages 85–100. Physica-Verlag HD, 2004.
10. J. Chen and A. Khalili. Order Selection in Finite Mixture Models With a Nonsmooth Penalty. *Journal of the American Statistical Association*, 103(484):1674–1683, 2008.
11. V. S. Cherkassky and F. Mulier. *Learning from Data: Concepts, Theory, and Methods*. John Wiley & Sons, Inc., New York, NY, USA, first edition, 1998.
12. Gene Ontology Consortium. The Gene Ontology project in 2008. *Nucleic Acids Research*, 36(Database-Issue):440–444, 2008.
13. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
14. E. Despierre, D. Lambrechts, P. Neven, F. Amant, S. Lambrechts, and I. Vergote. The molecular genetic basis of ovarian cancer and its roadmap towards a better treatment. *Gynecologic Oncology*, 117(2):358–365, 2010.
15. B. D'haene, J. Vandesompele, and J. Hellemans. Accurate and objective copy number profiling using real-time quantitative PCR. *Methods*, 50(4):262–270, 2010.
16. S. G. Durkin and T. W. Glover. Chromosome fragile sites. *Annual Review of Genetics*, 41(1):169–192, 2007.
17. M. A. T. Figueiredo and A. K. Jain. Unsupervised Learning of Finite Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, March 2002.
18. R. A. Fisher. On the interpretation of χ^2 from contingency tables, and the calculation of p. *Journal of the Royal Stat. Society*, 85(1):87–94, January 1922.
19. P. A. Futreal, L. Coin, M. Marshall, T. Down, T. Hubbard, R. Wooster, N. Rahman, and M. R. Stratton. A census of human cancer genes. *Nature reviews. Cancer*, 4(3):177–183, Mar 2004.
20. G. C. Garriga, E. Junttila, and H. Mannila. Banded structure in binary matrices. *Knowledge and Information Systems*, 28(1):197–226, 2011.
21. R. W. Hamming. *Numerical Methods for Scientists and Engineers*. Dover Publications, Inc., New York, NY, USA, second edition, 1986.
22. D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. Adaptive Computation and Machine Learning Series. MIT Press, 2001.
23. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, second edition, February 2009.
24. H. Z. Hausen. The search for infectious causes of human cancers: Where and why. *Virology*, 392(1):1–10, 2009.
25. J. Hollmén, J. K. Seppänen, and H. Mannila. Mixture models and frequent sets: combining global and local methods for 0-1 data. In *Proceedings of the Third SIAM International Conference on Data Mining*, pages 289–293.

Society of Industrial and Applied Mathematics, 2003.
26. J. Hollmén and J. Tikka. Compact and understandable descriptions of mixture of Bernoulli distributions. In M. R. Berthold, J. Shawe-Taylor, and N. Lavrač, editors, *Proceedings of the 7th International Symposium on Intelligent Data Analysis (IDA 2007)*, volume 4723 of *Lecture Notes in Computer Science*, pages 1–12, Ljubljana, Slovenia, September 2007. Springer-Verlag.
27. S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.
28. A. Hotho, S. Staab, and G. Stumme. Explaining text clustering results using semantic structures. In N. Lavrač, D. Gamberger, L. Todorovski, and H. Blockeel, editors, *Knowledge Discovery in Databases: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2003)*, volume 2838 of *LNAI*, pages 217–228. Springer, 2003.
29. T. Huang, H. Peng, and K. Zhang. Model Selection for Gaussian Mixture Models. *arXiv preprint arXiv:1301.3558*, 2013.
30. W. Hämäläinen. *Efficient search for statistically significant dependency rules in binary data*. PhD thesis, Department of Computer Science, University of Helsinki, Finland, 2010.
31. A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.
32. B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analyses. *SIAM Review*, 36(3):377–412, September 1994.
33. I. R. Kirsch. *The Causes and Consequences of Chromosomal Aberrations*. CRC Press, 1993.
34. W. Klösgen. Explora: a multipattern and multistrategy discovery assistant. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 249–271. American Association for Artificial Intelligence, 1996.
35. S. Knuutila, Y. Aalto, K. Autio, A. Björkqvist, W. El-Rifai, S. Hemmer, T. Huhta, E. Kettunen, S. Kiuru-Kuhlefelt, M.L. Larramendy, T Lushnikova, O. Monni, H. Pere, J. Tapper, M. Tarkkanen, A. Varis, V. Wasenius, M. Wolf, and Y. Zhu. DNA Copy Number Losses in Human Neoplasms. *Gynecologic Oncology*, 155(2):683–694, 1999.
36. I. Kononenko and M. Kukar. *Machine Learning and Data Mining: Introduction to Principles and Algorithms*. Horwood Publishing Limited, 2007.
37. M. Kuhn and K. Johnson. *Applied Predictive Modeling*. Springer, New York, Heidelberg, Dordrecht, London, 2013.
38. L. Langohr, V. Podpečan, M. Petek, I. Mozetič, K. Gruden, N. Lavrač, and H. Toivonen. Contrasting subgroup discovery. *The Computer Journal*, 56(3):289–303, 2013.
39. Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language*

Technologies-Volume 1, pages 1536–1545. Association for Computational Linguistics, 2011.
40. N. Lavrač, B. Kavšek, P. A. Flach, and L. Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188, 2004.
41. A. Lawrynowicz and J. Potoniec. Fr-ONT: an algorithm for frequent concept mining with formal ontologies. In M. Kryszkiewicz, H. Rybinski, A. Skowron, and Z. W. Raś, editors, *Foundations of Intelligent Systems, Proceedings of 19th International Symposium on Methodologies for Intelligent Systems (ISMIS 2011)*, volume 6804 of *Lecture Notes in Computer Science*, pages 428–437. Springer Berlin Heidelberg, 2011.
42. T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224–270, 1994.
43. B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data mining (KDD'98)*, pages 80–86. AAAI Press, August 1998.
44. D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova. Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research*, 33(Database issue):D54–D58, 2005.
45. G. J. McLachlan. On Bootstrapping the Likelihood Ratio Test Stastistic for the Number of Components in a Normal Mixture. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 36(3):318–324, 1987.
46. G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley series in probability and statistics. Wiley, second edition, 2008.
47. G. J. McLachlan and D. Peel. *Finite Mixture Models*, volume 299 of *Probability and Statistics – Applied Probability and Statistics Section*. Wiley, 2000.
48. Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 490–499. ACM, 2007.
49. V. Melnykov and R. Maitra. Finite mixture models and model–based clustering. *Statistics Surveys*, 4:80–116, 2010.
50. D. Mukherjee, Q. M. J. Wu, and T. M. Nguyen. Multiresolution based Gaussian mixture model for background suppression. *IEEE Transactions on Image Processing*, 22(12):5022–5035, 2013.
51. S. Myllykangas, J. Himberg, T. Böhling, B. Nagy, J. Hollmén, and S. Knuutila. DNA copy number amplification profiling of human neoplasms. *Oncogene*, 25(55):7324–7332, November 2006.
52. S. Myllykangas, J. Tikka, T. Böhling, S. Knuutila, and J. Hollmén. Classification of human cancers based on DNA copy number amplification modeling. *BMC Medical Genomics*, 1(15), May 2008.
53. P. Novak, N. Lavrač, and G. I. Webb. Supervised descriptive rule discovery: a unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research*, 10:377–403, February 2009.

54. G. Obe and Vijayalaxmi. *Chromosomal Alterations: Methods, Results, and Importance in Human Health*. Springer, 2007.
55. H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 27(1):29–34, 1999.
56. A. Oliveira-Brochado and F. V. Martins. Assessing the Number of Components in Mixture Models: a Review. FEP Working Papers 194, Universidade do Porto, Faculdade de Economia do Porto, November 2005.
57. P. Panov. *A Modular Ontology of Data Mining*. Doctoral dissertation, Jožef Stefan International Postgraduate School, July 2012.
58. G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In G. Piatetsky-Shapiro and W. J. Frawley, editors, *Knowledge Discovery in Databases*. AAAI/MIT Press, 1991.
59. J. R. Pollack, C. M. Perou, A. A. Alizadeh, M. B. Eisen, A. Pergamenschikov, C. F. Williams, S. S. Jeffrey, D. Botstein, and P. O. Brown. Genome-wide analysis of DNA copy-number changes using cDNA microarrays. *Nature Genetics*, 23(1):41–46, 1999.
60. P. M. V. Rancoita, M. Hutter, F. Bertoni, and I. Kwee. Bayesian DNA copy number analysis. *BMC Bioinformatics*, 10(1):10, 2009.
61. P. Ristoski and H. Paulheim. Feature Selection in Hierarchical Feature Spaces. In S. Džeroski, P. Panov, D. Kocev, and L. Todorovski, editors, *Discovery Science*, volume 8777 of *Lecture Notes in Computer Science*, pages 288–300. Springer International Publishing, 2014.
62. L. G. Shaffer and N. Tommerup. *ISCN 2005: An International System for Human Cytogenetic Nomenclature(2005) Recommendations of the International Standing Committee on Human Cytogenetic Nomenclature*. Karger, 2005.
63. P. Smyth. Model selection for probabilistic clustering using cross-validated likelihood. *Statistics and Computing*, 10(1):63–72, 2000.
64. A. Srinivasan. Aleph Manual, March 2007.
65. K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, 1981.
66. Sebastian Thrun. Extracting rules from artificial neural networks with distributed representations. In G. Tesauro, D.S. Touretzky, and T.K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 505–512. MIT Press, 1995.
67. J. Tikka, J. Hollmén, and S. Myllykangas. Mixture modeling of DNA copy number amplification patterns in cancer. In F. Sandoval, A. Prieto, J. Cabestany, and M. Graña, editors, *Proceedings of the 9th International Work-Conference on Artificial Neural Networks (IWANN 2007)*, volume 4507 of *Lecture Notes in Computer Science*, pages 972–979, San Sebastián, Spain, 2007. Springer-Verlag.
68. I. Trajkovski, N. Lavrač, and J. Tolar. SEGS: Search for enriched gene sets in microarray data. *Journal of Biomedical Informatics*, 41(4):588–601, 2008.

69. I. Trajkovski, F. Železný, N. Lavrač, and J. Tolar. Learning relational descriptions of differentially expressed gene groups. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 38(1):16–25, 2008.
70. Volker Tresp, Jürgen Hollatz, and Subutai Ahmad. Representing probabilistic rules with networks of gaussian basis functions. *Machine Learning*, 27(2):173–200, 1997.
71. E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1986.
72. A. Vavpetič and N. Lavrač. Semantic subgroup discovery systems and workflows in the SDM–toolkit. *The Computer Journal*, 56(3):304–320, 2013.
73. A. Vavpetič, P. K. Novak, M. Grčar, I. Mozetič, and N. Lavrač. Semantic data mining of financial news articles. In J. Fürnkranz, E. Hüllermeier, and T. Higuchi, editors, *Proceedings of Sixteenth International Conference on Discovery Science (DS 2013)*, volume 8140 of *Lecture Notes in Computer Science*, pages 294–307. Springer Berlin Heidelberg, 2013.
74. A. Vavpetič, V. Podpečan, and N. Lavrač. Semantic subgroup explanations. *Journal of Intelligent Information Systems*, 42(2):233–254, 2014.
75. B. Vogelstein and K.W. Kinzler. *The Genetic Basis of Human Cancer*. McGraw-Hill, New York, 2002.
76. U. Von Luxburg. *Clustering stability: An overview*. Now Publishers Inc, 2010.
77. E. Weinan. *Principles of Multiscale Modeling*. Cambridge University Press, 2011.
78. A. S. Willsky. Multiresolution markov models for signal and image processing. *In Proceedings of the IEEE*, 90(8):1396–1458, August 2002.
79. R. Wilson. MGMM: multiresolution Gaussian mixture models for computer vision. In *Proceedings of 15th International Conference on Pattern Recognition*, volume 1, pages 212–215, 2000.
80. M-J. Woo and T. N. Sriram. Robust Estimation of Mixture Complexity. *Journal of the American Statistical Association*, 101(476):1475–1486, December 2006.
81. S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proceedings of the First European Conference on Principles of Data Mining and Knowledge Discovery (PKDD '97)*, pages 78–87. Springer, 1997.
82. H.-M. Wu, Y.-J. Tien, and C.-H. Chen. GAP: A graphical environment for matrix visualization and cluster analysis. *Computational Statistics and Data Analysis*, 54(3):767–778, 2010.

5.3 Semantic Subgroup Discovery on Financial News Articles

Hedwig's initial application was in a financial domain with the goal to analyze financial news in search for interesting vocabulary patterns. This section describes the data acquisition process, methodology, and the experimental results.

5.3.1 Data acquisition and methodology

This section presents the results of applying the Hedwig system to get insight into a vast amount of news articles collected in a span of two years as part of the 7FP EU projects FIRST[1] and FOC[2]. We looked to get insight in the financial domain; more specifically, the vocabulary related to the European sovereign debt crisis used in news articles and financial blogs. We investigated the relationship between the financial market perception of a financial entity and the articles mentioning the financial entity. As a measure of market perception, we used the credit default swap (CDS) price. In essence, CDS is insurance for country bonds and reflects the market expectation that the issuer will default. The higher the CDS price, the more likely it is that that country will be unable to repay its debt [77].

Three sources of data were used: texts from news and blogs, CDS prices and a domain ontology. We started from a large database of annotated news articles (over 8 million), which were acquired using a data acquisition pipeline described in detail in the enclosed paper. We considered articles collected over eighteen months period from October 24, 2011 to January 13, 2013. Among other properties of each article (e.g., title and URL), the most important ones for our task are the information about which entities from a pre-defined European Sovereign Debt vocabulary appear in the given article (e.g., entities like "Portugal" or "Angela Merkel" or "austerity"). These entities (counting over 6,000) are part of a larger domain ontology which consists of several class hierarchies, e.g., the Euro crisis vocabulary, companies and banks, and geographical data.

We decided to focus our experiments on Portugal, as it is representative and was a financially troubled country in the analyzed period. Therefore the news articles were filtered to include only the ones mentioning Portugal.

The preparation stage consisted of two steps. The first step involved counting the number of times Portugal occurs together with every other entity of interest for each day of the collected history of articles. The second step involved selecting only the significant co-occurrences as example features. Each day represents one learning example and each example is described by the presence or absence of a certain entity that co-occurred with Portugal on that day. To filter out uninformative entities, we kept only the entities with a co-occurrence frequency at least 1.5 times greater than the average co-occurrence frequency over all days.

The target attribute for each example (one day) was computed from the CDS prices of Portugal and has three possible values that indicate the significant local extremes in the CDS price timelines: 'max' or 'min' if the local extreme was reached, respectively, or 'steady' if there was no change in the trend. These steps yielded a dataset of 337 examples, each with an average of 282 features (ranging between 35 and 761). The processed news and blogs articles, the CDS local extremes and the domain ontology were encoded as a set of RDF triples which were input to the Hedwig system.

The ontology that was used in the experiment has three main branches: financial entities, geographical entities and a specialized vocabulary of the European sovereign debt crisis. Some parts of the ontology were automatically induced by reusing various data

[1] http://project-first.eu/
[2] http://www.focproject.eu/

sources, while other parts, like the vocabulary, were constructed manually. Details are available in the enclosed paper at the end of this section.

5.3.2 Experimental results

The experimental data described in the previous section was used as input to the Hedwig semantic subgroup discovery system. We focused on finding subgroups for two target classes which represent trend reversals: the local maximum ('max') represents the date when the CDS price started to decrease and the local minimum ('min') the opposite. In both cases, we used the *WRAcc* subgroup discovery rule score, a beam width of 100, minimum coverage of 5 examples and the maximum number of predicates per rule of 6.

For the case of CDS price reaching the maximum (target class 'max'), the best scoring subgroup description was the following:

```
Max(X) ← reg_Western_Europe(X), Angela_Merkel(X), glo_austerity(X),
                    glo_recession(X). [28, 7]
```

For the case of CDS price reaching the minimum (target class 'min'), the best scoring subgroup description was the following:

```
Min(X) ← Index(X), comp_GALP_ENERGIA(X), Loan_Term(X),
                glo_fiscal_stimulus(X). [43, 8]
```

The first rule indicates that Portugal CDS prices reaching a local maximum are characterized by increased frequency of the following entities co-occurring with Portugal: the Western Europe region, Angela Merkel, and the terms 'austerity' and 'recession'. The numbers in brackets alongside the rules represent the total covered examples and the total positive covered examples, respectively. We should point out that a local maximum in a country's CDS price indicates that from that day on, the market expectation that the country will default decreased. Conversely, the second rule tells us that when the CDS prices reach a local minimum, we can notice an increased frequency of (stock) index terms, Portugal's corporation of natural and renewable energy companies (Galp Energia), loan terms and 'fiscal stimulus'. These results show that the higher the CDS prices, the more the sovereign debt vocabulary is used. When CDS prices are low, a more general financial terminology is used.

5.3.3 Related publication

Details of the methodology and experiments can be found in the following conference paper (included in this section):

A. Vavpetič, P. K. Novak, M. Grčar, I. Mozetič, and N. Lavrač, "Semantic data mining of financial news articles," in *Proceedings of Sixteenth International Conference on Discovery Science (DS 2013)*, J. Fürnkranz, E. Hüllermeier, and T. Higuchi, Eds., ser. Lecture Notes in Computer Science, vol. 8140, Springer Berlin Heidelberg, 2013, pp. 294–307, ISBN: 978-3-642-40896-0.

The author's contributions are as follows. Anže Vavpetič implemented the Hedwig system, preprocessed the data and executed the experiment. Petra Kralj Novak contributed to the experiment design and data preparation. Miha Grčar and Borut Sluban contributed the Dacq data acquisition and cleaning pipeline, while Igor Mozetič and Nada Lavrač contributed the idea of the experiment. All authors contributed to the text of the publication.

Semantic Data Mining of Financial News Articles

Anže Vavpetič[1,2], Petra Kralj Novak[1], Miha Grčar[1], Igor Mozetič[1], and Nada Lavrač[1,2,3]

[1] Jožef Stefan Institute, Ljubljana, Slovenia
[2] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
[3] University of Nova Gorica, Nova Gorica, Slovenia
anze.vavpetic@ijs.si

Abstract. Subgroup discovery aims at constructing symbolic rules that describe statistically interesting subsets of instances with a chosen property of interest. Semantic subgroup discovery extends standard subgroup discovery approaches by exploiting ontological concepts in rule construction. Compared to previously developed semantic data mining systems SDM-SEGS and SDM-Aleph, this paper presents a general purpose semantic subgroup discovery system Hedwig that takes as input the training examples encoded in RDF, and constructs relational rules by effective top-down search of ontologies, also encoded as RDF triples. The effectiveness of the system is demonstrated through an application in a financial domain with the goal to analyze financial news in search for interesting vocabulary patterns that reflect credit default swap (CDS) trend reversal for financially troubled countries. The approach is showcased by analyzing over 8 million news articles collected in the period of eighteen months. The paper exemplifies the results by showing rules reflecting interesting news topics characterizing Portugal CDS trend reversal in terms of conjunctions of terms describing concepts at different levels of the concept hierarchy.

Keywords: semantic data mining, subgroup discovery, ontology, credit default swap, financial crisis.

1 Introduction

This paper addresses the task of subgroup discovery, first defined by Klösgen [1] and Wrobel [2]. The goal of SD is to find subgroups of instances that are statistically interesting according to some property of interest for a given population of instances. SD is commonly described as being in the intersection of predictive and descriptive data mining as it is used for descriptive rule learning although the rules are induced from class-labeled data. Patterns discovered by subgroup discovery methods (called subgroup descriptions) are rules of the form `Class` ← `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 instances.

J. Fürnkranz, E. Hüllermeier, and T. Higuchi (Eds.): DS 2013, LNAI 8140, pp. 294–307, 2013.

It is well known from the literature on inductive logic programming (ILP) [3, 4] and relational data mining (RDM) [5] 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, are ontologies. Ontologies are consensually developed domain models that formally define the semantic descriptors and can act as means of providing additional information to machine learning (data mining) algorithms by attaching semantic descriptors to the data. Such domain knowledge is usually represented in a standard format which encourages knowledge reuse. Two popular formats are the Web Ontology Language (OWL) for ontologies and the Resource Description Framework (RDF) triplets for other structured data. 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 databases are now being translated into RDF as linked data. 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.

The process of exploiting formal ontologies within the process of data mining, called Semantic Data Mining (SDM), was formalized by Vavpetič and Lavrač [6]. Early work in using ontologies in machine learning and data mining is due to Kietz [7] who extended the standard learning bias used in ILP with description logic (DL) in his CLARIN-DL system. More recently, Lehmann and Haase [8] defined a refinement operator in a variant of DL, but considered only the construction of consistent and complete hypotheses. Lawrynowicz and Potoniec [9] introduced an algorithm for frequent concept mining in another variant of DL. Combining web mining and the semantic web was proposed by Berendt et al. [10]. Early work on this topic is due to Lisi et al. [11, 12], proposing an approach to mining the semantic web by using a hybrid language AL-log, used for mining multi-level association rules.

In this paper, we present a new semantic subgroup discovery system named Hedwig, which searches for subgroups with descriptions constructed from the given ontological vocabulary (including any provided binary relations). The traversal of the search space is effectively guided by the hierarchical structure of the ontology. The most relevant related work in exploiting ontologies in real-life data mining tasks is by Trajkovski et al. [13] who used the gene ontology to find enriched gene sets from microarray data, and by akova et al. [14] who used an ontology of Computer Aided Design elements and structures to find frequent design patterns.

In this paper, we present the results of applying the Hedwig system to get insight into a vast amount of news articles collected in last two years as part

of the 7FP EU projects FIRST and FOC. We seek for insight in the financial domain; more specifically we investigate the vocabulary related to the European sovereign debt crisis used in news articles and financial blogs. We investigate the relationship between the financial market perception of a financial entity and the articles mentioning the financial entity. As a measure of market perception, we use the credit default swap (CDS) price. In essence, CDS is insurance for country bonds and reflects the market expectation that the issuer will default. The higher the CDS price, the more likely it is that that country will be unable to repay its debt [15]. Portugal is the focus of our investigation as an example of a financially troubled country.

Gamberger et al. [16] employed SD techniques on a related problem. They have induced indicators of systemic banking crises by looking at past crises in the period 1976-2007. Rather than looking at news articles and relating them to the CDS prices, they used 105 publicly available financial indicators. Their main result is that demographic indicators are the most important: the percentage of the active population in connection to the annual percentage of money growth and the male life expectancy are especially crucial.

The main contributions of this paper are the new semantic data mining system named Hedwig, which is presented with its premiere application in understanding financial news, and the extensive data acquisition pipeline that was used for collecting the data. Another contribution is the first insights into the relationship between the European sovereign debt crisis vocabulary and the CDS price trends.

The paper is structured as follows. Section 2 describes the developed Hedwig semantic SD system. Section 3 describes the data acquisition and cleaning pipeline, while Section 4 describes the data preparation stage, the experimental setup and the results. Section 5 gives directions for further work and concludes the paper.

2 Methodology

This section describes the newly developed semantic subgroup discovery system Hedwig. Compared to standard subgroup discovery algorithms, Hedwig uses domain ontologies to guide the search space and formulate generalized hypothesis. Existing semantic subgroup discovery algorithms are either specialized for a specific domain [13] or adapted from systems that do not take into the account the hierarchical structure of background knowledge [6]. Hedwig overcomes these limitations as it is designed to be a general purpose semantic subgroup discovery system.

Semantic subgroup discovery, as addressed by the Hedwig system, results in relational descriptive rules, using training examples in RDF triples form and using several ontologies as background knowledge used. As an illustration, take three simplified ontologies illustrated in Figure 1, as sample ontologies which could be used in mining financial data.

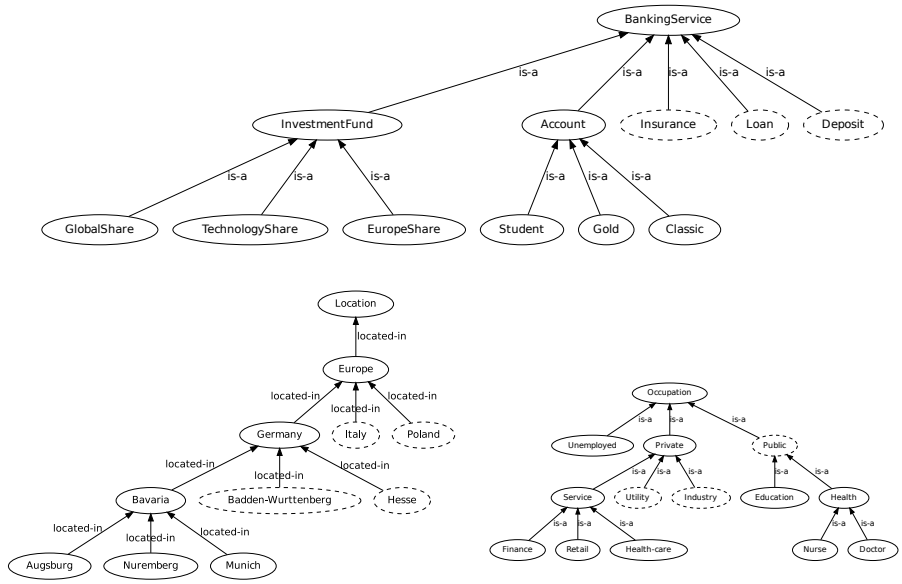


Fig. 1. The ontologies of banking services, locations and occupations. Concepts with omitted sub-concepts are drawn with a dashed line.

Formally, the semantic data mining task addressed in this paper is defined as follows.

Given:

- The empirical data in the form of a set of training examples expressed as RDF triples,
- Domain knowledge in the form of ontologies (one or more), and
- An object-to-ontology mapping which associates each object from the RDF triplets 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.

Subgroup describing rules are first-order logical expressions. Take the following rule, used to explain the format of induced subgroup describing rules.

```
Max(X) ← Country(X), Before(X,Y), comp_NESTLE_S_A(Y). [50, 10]
```

where variables `X`, `Y` represent sets of input instances. Note the convention that lowercase predicates (e.g., `comp_NESTLE_S_A`) represent specific instances (appearing in the leaves of the ontology), while capitalized predicates represent classes (appearing at higher hierarchy levels of the ontology), i.e., sets of specific instances (e.g., predicate `Country` subsumes instances like `cou_Portugal` or

298 A. Vavpetič et al.

```
function induce():
    rules = [default_rule]
    while improvement(rules):
        foreach rule in rules:
            rules.extend(specialize(rule))
        rules = best(rules, N)
    return rules

function specialize(rule):
    specializations = []
    foreach predicate in eligible(rule.predicates):
        # Specialize by traversing the subClassOf hierarchy
        for subclass in subclasses(predicate):
            new_rule = rule.swap(predicate, subclass)
            if can_specialize(new_rule):
                specializations = specializations.add(new_rule)
    if rule != default_rule:
        # Specialize by adding a new unary predicate to the rule
        new_predicate = next_non_ancestor(eligible(rule.predicates))
        new_rule = rule.append(new_predicate)
        if can_specialize(new_rule):
            specializations.add(new_rule)
    if rule.predicates.last().arity == 1:
        # Specialize by adding new binary predicates
        specializations.extend(add_binary_predicate(rule))
    return specializations
```

Fig. 2. Pseudo code of the Hedwig semantic SD algorithm

`cou_Slovenia`). The above rule is interpreted as follows. Let `Max(X)` denote a local maximum of credit default swap (CDS), which needs to be related with the information available in the extracted features of news articles at time point `X`. The countries `Country(X)`, which were frequently mentioned in articles on day `X` that is followed by `Y` in which the Nestle company was frequently mentioned. This rule condition is true for 50 input instances, 10 of which are of target class `Max`. The two numbers refer to coverage (the number of instances for which the rule body is true) and support (the number of instances for which both the rule head and body are true), respectively.

In order to search for interesting subgroups, we employed the algorithm described in Figure 2. The Hedwig system, which implements this algorithm, supports ontologies and examples to be loaded as a collection of RDF triples (a graph). The system automatically parses the RDF graph for the `subClassOf` hierarchy, as well as any other user-defined binary relations. Hedwig also defines a namespace of classes and relations for specifying the training examples to which the input must adhere.

The algorithm uses beam search, where the beam contains the best N rules found so far. The search starts with the default rule which covers all input examples. In every iteration of the search, each rule from the beam is specialized via one of the three operations:

1. Replace the rules predicate with a predicate that is a sub-class of the previous one, e.g., `City(X)` is specialized to `Capital(X)`.
2. Append a new unary predicate to the rule, e.g., `Max(X)` ← `City(X)` is specialized to `Max(X)` ← `City(X), Company(X)`.
3. Append a new binary predicate, thus introducing a new existentially quantified variable, e.g.: `Max(X)` ← `City(X)` is specialized to `Max(X)` ← `City(X), Before(X,Y)`.[1]

Rule induction via specializations is a well-established way of inducing rules, since every specialization either maintains or reduces the current number of covered examples. A rule will not be specialized once its coverage is zero or falls below some predetermined threshold. After the specialization step is applied to each rule in the beam, a new selection of the best scoring N rules is made. If no improvement is made to the collection of rules, the search is stopped. In principle, our procedure supports any rule scoring function. Currently we implemented the popular SD scoring functions WRAcc [17], χ^2 for discrete target classes [18], and Z-score for ranked examples [19].

3 Data Acquisition and Cleaning

In this section, we present the data acquisition pipeline by describing each of its components.

The pipeline consists of several technologies that interoperate to achieve the desired goal, i.e., preparing the data for further analysis. It is responsible for acquiring unstructured data from several data sources, preparing it for the analysis, and brokering it to the appropriate analytical components. Our data acquisition pipeline is running continuously (since October 24, 2011), polling the Web and proprietary APIs for recent content, turning it into a stream of preprocessed text documents.

The news articles and web blogs are collected from 175 web sites and 2,600 RSS feeds, intentionally selected to have a strong bias for finance. We collect data from the main news providers and aggregators (like yahoo.com, dailymail.co.uk, nytimes.com, bbc.co.uk, wsj.com) and also from the main financial blogs (like zerohedge.com). The hundred most productive web sites account for 85% of collected documents. The fifty most productive domains with their average document production per day are displayed in Figure 3.

In the period from October 24, 2011 to March 31, 2013, 8,703,895 documents were collected and processed. On an average work day, about 18,000 articles are

[1] Note that variable `Y` needs to be 'consumed' by a literal to be conjunctively added to this clause in the next step of rule refinement.

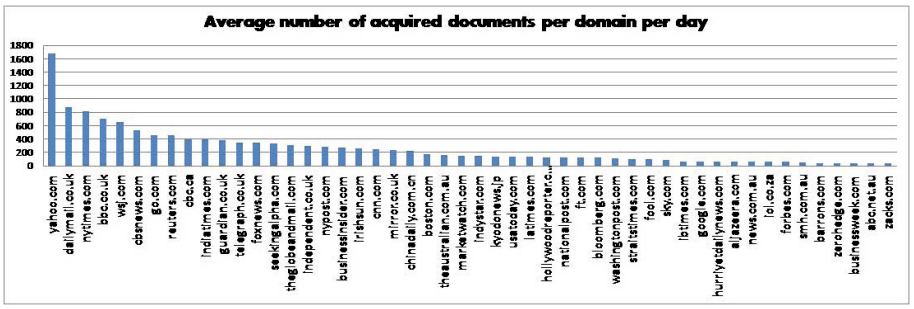


Fig. 3. The average number of acquired documents per domain per day for the fifty most productive domains. The hundred most productive web sites account for 85% of our acquired documents.

collected. The number of collected articles is substantially lower during weekends; around 10,000 per weekend day. Holidays are also characterized by a lower number of documents. The number of collected documents per day is presented in Figure 4.

When dealing with official news streams, some pre-processing steps can be avoid-ed. Official news is provided in a semi-structured fashion such that titles, publication dates, and other metadata are clearly indicated. Furthermore, named entities (i.e., company names and stock symbols) are identified in texts and article bodies are provided in a raw textual format without any boilerplate (i.e., undesired content such as advertisements, copyright notices, navigation elements, and recommendations).

Content from blogs, forums, and other Web content, however, is not immediately ready to be processed by the text analysis methods. Web pages contain a lot of noise that needs to be identified and removed before the content can be analyzed. For this reason, we have developed DacqPipe (or Dacq), a data acquisition and pre-processing pipeline. Dacq consists of (i) data acquisition components, (ii) data clean-ing components, (iii) natural-language preprocessing components, (iv) semantic anno-tation components, and (v) ZeroMQ emitter components.

The data acquisition components are mainly RSS readers that poll for data in parallel. One RSS reader is instantiated for each Web site of interest. The RSS sources, corresponding to a particular Web site, are polled one after another by the same RSS reader to prevent the servers from rejecting requests due to concurrency. An RSS reader, after it has collected a new set of documents from an RSS source, dispatches the data to one of several processing pipelines. The pipeline is chosen according to its current load size (load balancing). A processing pipeline consists of a boilerplate remover, duplicate detector, language detector, sentence splitter, tokenizer, part-of-speech tagger, lemmatizer, stop-word detector and a semantic annotator. Some of the components are custom-made while other use the functionality available from the OpenNLP library . Each pipeline component is described in more detail below.

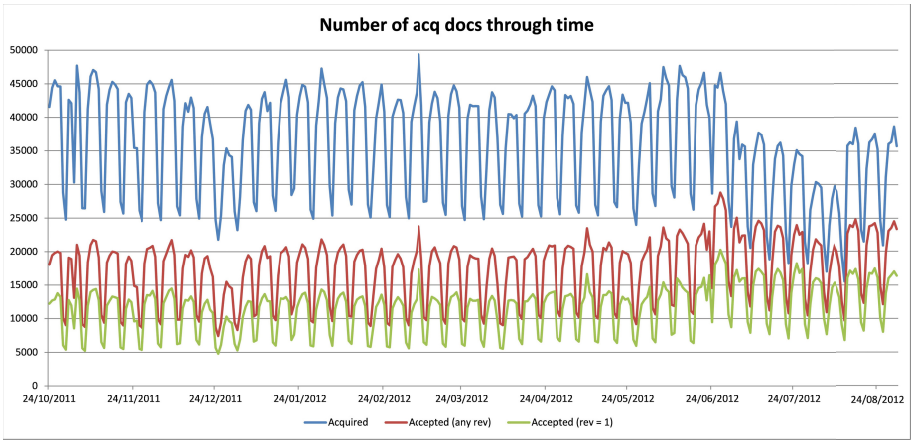


Fig. 4. The number of acquired documents per day. The top line represents the number of all acquired documents. The bottom line represents the documents that our system sees for the first time and the middle line represents the revisions of already acquired documents.

- *Boilerplate Remover*. Extracting meaningful content from Web pages presents a challenging problem. Our setting focuses on content extraction from streams of HTML documents. The developed infrastructure converts continuously acquired HTML documents into a stream of plain text documents. Our novel content extraction algorithm is efficient, unsupervised, and language-independent. The information extraction approach is based on the observation that HTML documents from the same source normally share a common template. The core of the proposed content extraction algorithm is a simple data structure called URL Tree. The performance of the algorithm was evaluated in a stream setting on a time-stamped semi-automatically annotated dataset which was made publicly available.
- *Duplicate Detector*. News aggregators are websites that aggregate web content such as news articles in one location for easy viewing. They cause articles to appear on the web with many different URLs pointing to it. To have a concise dataset of unique articles, we developed a duplicate detector that is able to see if the document was already acquired or not.
- *Language Detector*. By using a machine learning model, it detects the language and discards all the documents that are detected to be non-English. The model is trained on a large multilingual set of documents. The basic features for the model are frequencies of two consecutive words.
- *Sentence Splitter*. Splits the text into sentences. The result is the input to the part-of-speech tagger. We use the OpenNLP implementation of the Sentence splitter.

- *Tokenizer.* Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. In our pipeline, we use our own implementation of the tokenizer, which supports the Unicode character set and is based on rules.
- *Part-of-speech Tagger.* The Part of Speech (POS) Tagger marks tokens with their corresponding word type (e.g., noun, verb, proposition) based on the token itself and the context of the token. A token might have multiple POS tags depending on the token and the context. The part-of-speech tagger from the OpenNLP library is used.
- *Lemmatizer.* Lemmatization is the process of finding the normalized forms of words appearing in text. It is a useful preprocessing step for a number of language engineering and text mining tasks, and especially important for languages with rich inflectional morphology. In our data acquisition pipeline, we use LemmaGen [20] for lemmatization, which is the most efficient publicly available lemmatizer trained on large lexicons of multiple languages, whose learning engine can be retrained to effectively generate lemmatizers of other languages. We lemmatize to English.
- *Stop-word detector.* In automated text processing, stop words are words that do not carry semantic meaning. In our data acquisition pipeline, stop words are detected and annotated.
- *Semantic annotator.* Each entity has associated gazetteers; gazetteers are rules describing the entity in text. For example, "The United States of America" can appear in text as "USA", "US", "The United States", and so on. The rules include capitalization, lemmatization, POS tag constraints, must-contain constraints (another gazetteer must be detected in the document or in the sentence) and followed-by constraints.

4 Financial Use Case

First, this section presents the data and the data preparation stage needed to apply the proposed methodology. Three sources of data were used: texts from news and blogs, CDS prices and a domain ontology. Finally, this section present the experimental results achieved by applying subgroup discovery on the prepared data.

We started from a large database of annotated news articles (over 8 million), which were acquired using the data acquisition pipeline presented in the previous section. We considered articles collected over eighteen months period from October 24, 2011 to January 13, 2013. Among other properties of each article (e.g., title and URL), the most important ones for our task are the information about which entities from a pre-defined European Sovereign Debt vocabulary appear in the given article (e.g., entities like "Portugal" or "Angela Merkel" or "austerity"). These entities (counting over 6,000) are part of a larger domain ontology which consists of several class hierarchies, e.g., the Euro crisis vocabulary, companies and banks, and geographical data.

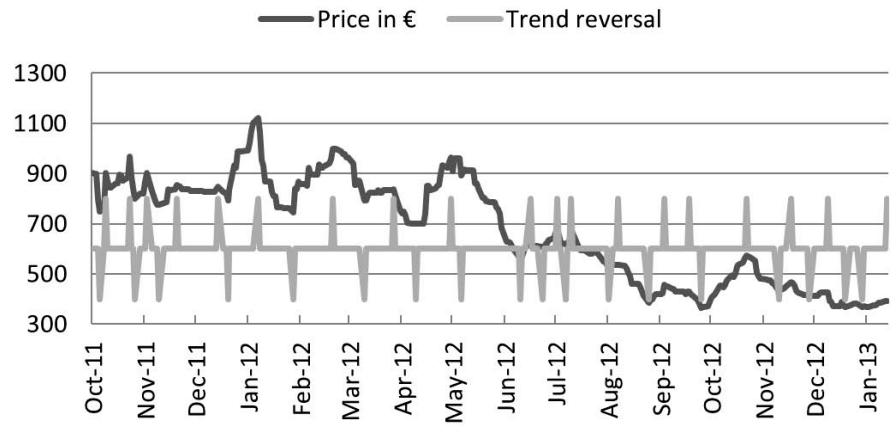


Fig. 5. Portugal CDS prices and trend reversals between October 2011 and January 2013. Upward spikes indicate local maxima, while downward spikes indicate local minima.

We decided to focus our experiments on Portugal, as it is representative and was a financially troubled country in the analyzed period. Therefore the news articles were filtered to include only the articles mentioning Portugal. The preparation stage consisted of two steps. The first step involved counting the number of times Portugal occurs together with every other entity of interest for each day of the collected history of articles. The second step involved selecting only the significant co-occurrences as example features. Each day represents one learning example and each example is described by the presence or absence of a certain entity that co-occurred with Portugal on that day. To filter out uninformative entities, we kept only the entities with a co-occurrence frequency at least 1.5 times greater than the average co-occurrence frequency over all days.

The target attribute for each example (day) was computed from the CDS prices of Portugal and has three possible values that indicate the significant local extremes in the CDS price timelines: 'max' or 'min' if the local extreme was reached, respectively, or 'steady' if there was no change in the trend (Figure 5). These steps yielded a dataset of 337 examples, each with an average of 282 features (ranging between 35 and 761).

The processed news and blogs articles, the CDS local extremes and the domain ontology were encoded as a set of RDF triples which were input to the Hedwig system.

The financial ontology which we actually used in the experiments is illustrated in Figure 6. The ontology has three main branches: financial entities, geographical entities and a specialized vocabulary of the European sovereign debt crisis. Some parts of the ontology were automatically induced by reusing various data sources, while other parts, like the vocabulary, were constructed manually.

Each entity in the ontology is equipped with a gazetteer. The gazetteer contains lexical knowledge about the possible forms in which the entity occurs in texts. This knowledge is used by the entity recognition engine which is attached to the data acquisition pipeline. Note that the gazetteers are initially built automatically in the ontology construction process. This approach to entity recognition is prone to errors due to homographs, i.e., words that are spelled the same but have different meanings. This is especially prominent for acronyms and stock symbols. To improve the entity recognition process and to reduce the noise in the stream of discovered entities, we have performed several semi-automated ontology refinement iterations.

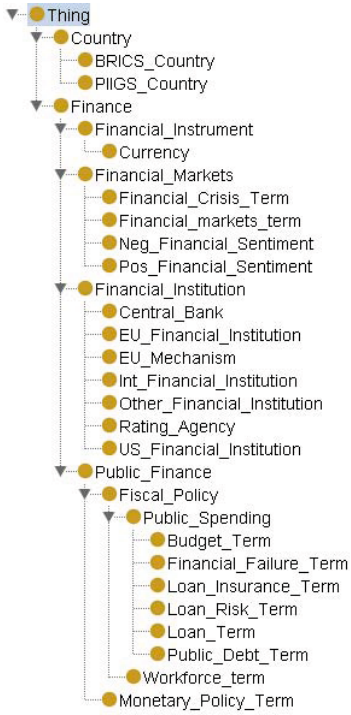


Fig. 6. The ontology that conceptualizes the European financial crisis vocabulary

We used the IDMS database and MSN Money[2] to grow the ontology from a list of seed stock indices to its constituents (stocks) and further on to the companies that issue these stocks. This resulted in to 2019 financial entities (like banks, companies, investment funds, stocks and stock indexes). The geographical part of the ontology was generated from GeoNames[3] (countries, cities, regions, etc). We selected 598 most important geographical entities and included them into the ontology. The specialized vocabulary of the European financial crisis (166 terms) was developed manually by using expert knowledge (Figure 6). The main protagonists of the crisis were taken from Wikipedia[4].

In our experiment, we focused on finding subgroups for two target classes which represent trend reversals: the local maximum ('max') represents the date when the CDS price started to decrease and the local minimum ('min') the opposite. In both cases, we used the WRAcc subgroup discovery rule score, a beam width of 100, minimum coverage of 5 examples and the maximum number of predicates per rule of 6.

[2] http://money.msn.com/
[3] http://www.geonames.org/
[4] http://en.wikipedia.org/wiki/List_of_protagonists:_European_sovereign-debt_crisis

For the case of CDS price reaching the maximum (target class 'max'), the best scoring subgroup was:

```
Max(X) ← reg_Western_Europe(X), Angela_Merkel(X),
   glo_austerity(X), glo_recession(X). [28, 7]
```

For the case of CDS price reaching the minimum (target class 'min'), the best scoring subgroup was:

```
Min(X) ← Index(X), comp_GALP_ENERGIA(X), Loan_Term(X),
            glo_fiscal_stimulus(X). [43, 8]
```

The first rule indicates that Portugal CDS prices reaching a local maximum are characterized by increased frequency of the following entities co-occurring with Portugal: the Western Europe region, Angela Merkel, and the terms 'austerity' and 'recession'. We should point out that a local maximum in a country's CDS price indicates that from that day on, the market expectation that the country will default decreased. Conversely, the second rule tells us that when the CDS price reach a local minimum, we can notice an increased frequency of (stock) index terms, Portugal's corporation of natural and renewable energy companies (Galp Energia), loan terms and 'fiscal stimulus'. These results show that the higher the CDS prices, the more the sovereign debt vocabulary is used. When CDS prices are low, a more general financial terminology is used.

5 Conclusions

The newly developed semantic subgroup discovery system Hedwig was presented, which overcomes the limitations of existing semantic subgroup discovery systems. Compared to standard subgroup discovery, novelties of this paper are the exploitation of the ontology to generalize over the entities, while also using of the user-provided binary relations and using the `subClassOf` relation to guide the search procedure. We are currently performing a comprehensive study which should result in a comparison of the new system with the related work.

We employed Hedwig for analyzing news articles about Portugal during the last year and a half. Using co-occurrence frequencies of entities appearing together with Portugal, a domain ontology linking the entities into a formal hierarchy, and a history of Credit Default Swap (CDS) prices, we induced subgroups describing prominent entities appearing at times of CDS trend reversals (either upward or downward). The extracted subgroup descriptions give us a clear indication that news articles content indeed reflects the CDS prices. Having this information, we are encouraged to proceed with building a model for CDS trend reversal prediction. For this purpose, we plan to include additional information about the entities (e.g., TF-IDF weights) and extra-textual information (not only the pre-defined ontological entities) into the input data. Additionally, we will employ several classification algorithms and compare them.

Acknowledgments. This work was supported by the Slovenian Research Agency [grants P-103 and P-04431] and the EU projects "Large scale information extraction and integration infrastructure for supporting financial decision making" (FIRST, grant agreement 257928) and "Forecasting Financial Crises" (FOC, grant agreement 255987).

References

[1] Klösgen, W.: Explora: a multipattern and multistrategy discovery assistant. In: Advances in Knowledge Discovery and Data Mining, pp. 249–271. American Association for Artificial Intelligence, Menlo Park (1996)
[2] Wrobel, S.: An algorithm for multi-relational discovery of subgroups. In: Komorowski, J., Żytkow, J.M. (eds.) PKDD 1997. LNCS, vol. 1263, pp. 78–87. Springer, Heidelberg (1997)
[3] Muggleton, S. (ed.): Inductive Logic Programming. The APIC Series, vol. 38. Academic Press (1992)
[4] De Raedt, L.: Logical and Relational Learning. Springer, Heidelberg (2008)
[5] Džeroski, S., Lavrač, N. (eds.): Relational Data Mining. Springer, Berlin (2001)
[6] Vavpetič, A., Lavrač, N.: Semantic subgroup discovery systems and workflows in the SDM-Toolkit. Comput. J. 56(3), 304–320 (2013)
[7] Kietz, J.-U.: Learnability of description logic programs. In: Matwin, S., Sammut, C. (eds.) ILP 2002. LNCS (LNAI), vol. 2583, pp. 117–132. Springer, Heidelberg (2003)
[8] Lehmann, J., Haase, C.: Ideal downward refinement in the $\mathcal{EL}$ description logic. In: De Raedt, L. (ed.) ILP 2009. LNCS, vol. 5989, pp. 73–87. Springer, Heidelberg (2010)
[9] Ławrynowicz, A., Potoniec, J.: Fr-ONT: An algorithm for frequent concept mining with formal ontologies. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS, vol. 6804, pp. 428–437. Springer, Heidelberg (2011)
[10] Berendt, B., Hotho, A., Stumme, G.: Towards semantic web mining. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 264–278. Springer, Heidelberg (2002)
[11] Lisi, F.A., Malerba, D.: Inducing multi-level association rules from multiple relations. Machine Learning 55, 175–210 (2004), 10.1023/B:MACH.0000023151.65011.a3
[12] Lisi, F.A., Esposito, F.: Mining the semantic web: A logic-based methodology. In: Hacid, M.-S., Murray, N.V., Raś, Z.W., Tsumoto, S. (eds.) ISMIS 2005. LNCS (LNAI), vol. 3488, pp. 102–111. Springer, Heidelberg (2005)
[13] Trajkovski, I., Železný, F., Lavrač, N., Tolar, J.: Learning relational descriptions of differentially expressed gene groups. IEEE Transactions on Systems, Man, and Cybernetics, Part C 38(1), 16–25 (2008)
[14] Žáková, M., Železný, F., Garcia-Sedano, J.A., Tissot, C.M., Lavrač, N., Křemen, P., Molina, J.: Relational data mining applied to virtual engineering of product designs. In: Muggleton, S.H., Otero, R., Tamaddoni-Nezhad, A. (eds.) ILP 2006. LNCS (LNAI), vol. 4455, pp. 439–453. Springer, Heidelberg (2007)
[15] Hull, J., Predescu-Vasvari, M., White, A., Rotman, J.L.: The relationship between credit default swap spreads, bond yields, and credit rating announcements (2002)

[16] Gamberger, D., Lučanin, D., Šmuc, T.: Descriptive modeling of systemic banking crises. In: Ganascia, J.-G., Lenca, P., Petit, J.-M. (eds.) DS 2012. LNCS, vol. 7569, pp. 67–80. Springer, Heidelberg (2012)
[17] Lavrač, N., Kavšek, B., Flach, P., Todorovski, L.: Subgroup discovery with CN2-SD. Journal of Machine Learning Research 5, 153–188 (2004)
[18] Shimada, K., Hirasawa, K., Hu, J.: Class association rule mining with chi-squared test using genetic network programming. In: IEEE International Conference on Systems, Man and Cybernetics, SMC 2006, vol. 6, pp. 5338–5344 (2006)
[19] DeGroot, M.H., Schervish, M.J.: Probability and Statistics, ch. 8, 9. Addison-Wesley (2002)
[20] Juršič, M., Mozetič, I., Erjavec, T., Lavrač, N.: Lemmagen: Multilingual lemmatisation with induced ripple-down rules. J. UCS 16(9), 1190–1214 (2010)

Chapter 6

Conclusions and Further Work

This thesis presents a formal framework for semantic subgroup discovery and several new algorithms, which were applied in multiple scenarios. It also significantly contributes to the field of relational data mining. We dedicated a chapter to describing the RDM task, the propositionalization technique, and to our contributions in the form of a RDM package for Python and a set of components for the data mining platform ClowdFlows.

A central contribution of this thesis is the semantic subgroup discovery (SSD) formal framework, and three original SSD implementations: SDM-SEGS, SDM-Aleph and Hedwig. We evaluated the approaches on several experimental datasets and three main applications: explaining subgroups of breast cancer patients, multi-resolution 0-1 data analysis, and financial news article analysis. Lastly, all of the developed software is open-source and available for use as libraries or as components in the ClowdFlows data mining platform.

We envision several directions of further research, some of which we have already started working on. We divide the further work into SDM scalability and improvements to the RDM package.

Scaling SDM using network analysis The utility of SDM on large datasets is currently limited at around 10^4 to 10^5 graph nodes. Given the availability of huge Linked Open datasets such as Bio2RDF[1], containing several billion connections, experimentation with such knowledge bases is presently infeasible.

An interesting approach would be to employ scalable network analysis techniques (such as community detection) to detect network parts relevant for the experiment at hand. If we used only this relevant sub-network in combination with SDM, we anticipate that the resulting models would be comparably accurate and computed much faster, given the smaller search space. The work in this direction has already started and shows promising results.

SDM parellelization Like rule induction, Hedwig and other SSD algorithms, are not trivially parallelizable. This means that in order to parallelize the approach, we need to make fundamental changes to the algorithm, potentially leading to different results (i.e., not necessarily worse results). There already exists a number of existing potentially applicable parallelization or distributed data mining techniques for classification rule and tree induction. One such approach could work as follows. We have n available processing units (e.g., stand-alone computers or cores), where each unit receives a subsample of the data. Each unit derives its own local model (e.g., a set of rules). During this phase the units can optionally communicate. A combining procedure merges all of the local models

[1] http://bio2rdf.org/

into a single model. For example, a simple combining procedure would be to evaluate the score of each local rule on the complete dataset and to keep only the best rules.

The described approach distributes the data and the basic algorithm remains intact. In contrast, we can think of ways of parallelizing the rule constructing algorithm itself, i.e., adapting it so that the workload can be divided among several processing units. For example, inspired by synchronous tree construction we could divide the background knowledge among units, so that each unit determines the best specializations only for its own problem subspace. The units would then communicate the best possible rule specializations among each other. This also means that each unit would need to hold the complete model in its memory, but not the complete background knowledge.

RDM package improvements Regarding the RDM package we envision various possible improvements. Mainly, adding more approaches and new preprocessing and visualization components into ClowdFlows. More specifically, we plan to add Statistical Relational Learning approaches, as well as other popular ILP/RDM approaches (e.g., FOIL and Progol), that are currently missing from the package.

Currently, the ClowdFlows package contains very limited support for preprocessing relational datasets. We plan to implement tools for attribute selection, data filtering, and data and model visualizations. Preferably, these should be agnostic to the type of data—either propositional or relational—to be usable in all ClowdFlows scenarios.

References

[1] S. Džeroski and P. A. Flach, Eds., *Inductive Logic Programming, 9th International Workshop, ILP-99, Bled, Slovenia, June 24-27, 1999, Proceedings*, vol. 1634, ser. Lecture Notes in Computer Science, Springer, 1999, ISBN: 3-540-66109-3.

[2] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Magazine*, vol. 17, no. 3, pp. 37–54, 1996. [Online]. Available: http://www.aaai.org/ojs/index.php/aimagazine/article/view/1230.

[3] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986. DOI: 10.1023/A:1022643204877. [Online]. Available: http://dx.doi.org/10.1023/A:1022643204877.

[4] ——, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993, ISBN: 1-55860-238-0.

[5] P. Clark and T. Niblett, "The CN2 induction algorithm," *Machine Learning*, vol. 3, pp. 261–283, 1989. DOI: 10.1007/BF00116835. [Online]. Available: http://dx.doi.org/10.1007/BF00116835.

[6] W. W. Cohen, "Fast effective rule induction," in *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, A. Prieditis and S. J. Russell, Eds., Morgan Kaufmann, 1995, pp. 115–123, ISBN: 1-55860-377-8.

[7] J. Fürnkranz, D. Gamberger, and N. Lavrač, *Foundations of Rule Learning*, ser. Cognitive Technologies. Springer, 2012, ISBN: 978-3-540-75196-0. DOI: 10.1007/978-3-540-75197-7. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-75197-7.

[8] P. K. Novak, N. Lavrač, and G. I. Webb, "Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining," *Journal of Machine Learning Research*, vol. 10, pp. 377–403, 2009. DOI: 10.1145/1577069.1577083. [Online]. Available: http://doi.acm.org/10.1145/1577069.1577083.

[9] R. S. Michalski, "A theory and methodology of inductive learning," *Artif. Intell.*, vol. 20, no. 2, pp. 111–161, 1983. DOI: 10.1016/0004-3702(83)90016-4. [Online]. Available: http://dx.doi.org/10.1016/0004-3702(83)90016-4.

[10] R. S. Michalski, I. Mozetič, J. Hong, and N. Lavrač, "The multi-purpose incremental learning system AQ15 and its testing application to three medical domains," in *Proceedings of the 5th National Conference on Artificial Intelligence. Philadelphia, PA, August 11-15, 1986. Volume 2: Engineering.*, T. Kehler and S. J. Rosenschein, Eds., Morgan Kaufmann, 1986, pp. 1041–1047. [Online]. Available: http://www.aaai.org/Library/AAAI/1986/aaai86-171.php.

[11] S. Džeroski and N. Lavrač, Eds., *Relational Data Mining*. Springer, Sep. 2001, ISBN: 3-540-42289-7.

[12] R. A. Kowalski, "Algorithm = logic + control," *Commun. ACM*, vol. 22, no. 7, pp. 424–436, 1979. DOI: 10.1145/359131.359136. [Online]. Available: http://doi.acm.org/10.1145/359131.359136.

[13] S. Muggleton, Ed., *Inductive Logic Programming*. Academic Press, London, 1992.

[14] L. De Raedt, *Logical and Relational Learning*, L. D. Raedt, Ed. Springer, 2008, ISBN: 978-3-540-68856-3.

[15] N. Guarino, D. Oberle, and S. Staab, "What is an ontology?" In *Handbook on Ontologies*, ser. International Handbooks on Information Systems, S. Staab and R. Studer, Eds., Springer, 2009, pp. 1–17, ISBN: 978-3-540-70999-2. DOI: 10.1007/978-3-540-92673-3_0. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-92673-3_0.

[16] N. Lavrač, A. Vavpetič, L. Soldatova, I. Trajkovski, and P. Kralj Novak, "Using ontologies in semantic data mining with SEGS and g-SEGS," in *Proceedings of the International Conference on Discovery Science (DS '11)*, Springer, 2011, pp. 165–178.

[17] A. Vavpetič and N. Lavrač, "Semantic subgroup discovery systems and workflows in the SDM-toolkit," *The Computer Journal*, vol. 56, no. 3, pp. 304–320, 2013 (included in Chapter 4 of this thesis).

[18] I. Trajkovski, F. Železný, N. Lavrač, and J. Tolar, "Learning relational descriptions of differentially expressed gene groups," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 38, no. 1, pp. 16–25, 2008.

[19] M. Perovšek, A. Vavpetič, J. Kranjc, B. Cestnik, and N. Lavrač, "Wordification: Propositionalization by unfolding relational data into bags of words," *Expert Syst. Appl.*, vol. 42, no. 17-18, pp. 6442–6456, 2015 (not included in this thesis).

[20] N. Lavrač, M. Perovšek, and A. Vavpetič, "Propositionalization online," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part III*, T. Calders, F. Esposito, E. Hüllermeier, and R. Meo, Eds., ser. Lecture Notes in Computer Science, vol. 8726, Springer, 2014 (included in Chapter 3 of this thesis), pp. 456–459, ISBN: 978-3-662-44844-1. DOI: 10.1007/978-3-662-44845-8. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-44845-8.

[21] A. Vavpetič and N. Lavrač, "Semantic subgroup discovery systems and workflows in the SDM-toolkit," *The Computer Journal*, vol. 56, no. 3, pp. 304–320, 2013.

[22] A. Vavpetič, P. K. Novak, and N. Lavrač, "Analysing financial vocabulary using a new semantic subgroup discovery system hedwig," in *Proceedings of the 5th Jožef Stefan International Postgraduate School Students Conference*, Ljubljana, Slovenia, 23 May 2013, pp. 219–229.

[23] A. Vavpetič, P. K. Novak, M. Grčar, I. Mozetič, and N. Lavrač, "Semantic data mining of financial news articles," in *Proceedings of Sixteenth International Conference on Discovery Science (DS 2013)*, J. Fürnkranz, E. Hüllermeier, and T. Higuchi, Eds., ser. Lecture Notes in Computer Science, vol. 8140, Springer Berlin Heidelberg, 2013, pp. 294–307, ISBN: 978-3-642-40896-0.

[24] P. R. Adhikari, A. Vavpetič, J. Kralj, N. Lavrač, and J. Hollmén, "Explaining mixture models through semantic pattern mining and banded matrix visualization," *Machine Learning Journal*, in press 2016.

[25] A. Vavpetič, V. Podpečan, S. Meganck, and N. Lavrač, "Explaining subgroups through ontologies," in *PRICAI 2012: Trends in Artificial Intelligence - 12th Pacific Rim International Conference on Artificial Intelligence, Kuching, Malaysia, September 3-7, 2012. Proceedings*, P. Anthony, M. Ishizuka, and D. Lukose, Eds., ser. Lecture Notes in Computer Science, vol. 7458, Springer, 2012, pp. 625–636, ISBN: 978-3-642-32694-3. DOI: 10.1007/978-3-642-32695-0. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-32695-0.

[26] A. Vavpetič, V. Podpečan, and N. Lavrač, "Semantic subgroup explanations," *J. Intell. Inf. Syst.*, vol. 42, no. 2, pp. 233–254, 2014.

[27] P. R. Adhikari, A. Vavpetič, J. Kralj, N. Lavrač, and J. Hollmén, "Explaining mixture models through semantic pattern mining and banded matrix visualization," in *Discovery Science - 17th International Conference, DS 2014, Bled, Slovenia, October 8-10, 2014. Proceedings*, S. Džeroski, P. Panov, D. Kocev, and L. Todorovski, Eds., ser. Lecture Notes in Computer Science, vol. 8777, Springer, 2014, pp. 1–12, ISBN: 978-3-319-11811-6. DOI: 10.1007/978-3-319-11812-3. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-11812-3.

[28] A. Vavpetič, P. K. Novak, M. Grčar, I. Mozetič, and N. Lavrač, "Semantic data mining of financial news articles," in *Proceedings of Sixteenth International Conference on Discovery Science (DS 2013)*, J. Fürnkranz, E. Hüllermeier, and T. Higuchi, Eds., ser. Lecture Notes in Computer Science, vol. 8140, Springer Berlin Heidelberg, 2013, pp. 294–307, ISBN: 978-3-642-40896-0 (included in Chapter 5 of this thesis).

[29] W. Klösgen, "Advances in knowledge discovery and data mining," in, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1996, ch. Explora: a multipattern and multistrategy discovery assistant, pp. 249–271, ISBN: 0-262-56097-6. [Online]. Available: http://dl.acm.org/citation.cfm?id=257938.257965.

[30] S. Wrobel, "An algorithm for multi-relational discovery of subgroups," in *Proceedings of the 1st European Conference on Principles of Data Mining and Knowledge Discovery (PKDD'97)*, Trondheim, Norway: Springer-Verlag, Berlin, Germany, 24-27 June 1997, pp. 78–87.

[31] A. Srinivasan, *Aleph manual*, http://www.cs.ox.ac.uk/activities/machinelearning/Aleph/, Mar. 2007.

[32] T. Hofweber, "Logic and ontology," in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., Fall 2014, 2014.

[33] B. Smith, "Blackwell guide to the philosophy of computing and information," in, L. Floridi, Ed. Oxford Blackwell, 2003, ch. Ontology, pp. 155–166.

[34] M. Krötzsch, F. Simancik, and I. Horrocks, "A description logic primer," *CoRR*, vol. abs/1201.4089, 2012. [Online]. Available: http://arxiv.org/abs/1201.4089.

[35] F. Baader, "Description logic terminology," in *The Description Logic Handbook: Theory, Implementation, and Applications*, F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, Eds., Cambridge University Press, 2003, pp. 485–495, ISBN: 0-521-78176-0.

[36] R. Srikant and R. Agrawal, "Mining generalized association rules," in *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases, September 11-15, 1995, Zurich, Switzerland.*, U. Dayal, P. M. D. Gray, and S. Nishio, Eds., Morgan Kaufmann, 1995, pp. 407–419, ISBN: 1-55860-379-4. [Online]. Available: http://www.vldb.org/conf/1995/P407.PDF.

[37] R. M. J. Ayres and M. T. P. Santos, "Ontgar algorithm: An ontology-based algorithm for mining generalized association rules," in *9th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2012, 29-31 May 2012, Chongqing, China*, IEEE, 2012, pp. 656–660. DOI: 10.1109/FSKD.2012.6233861. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/FSKD.2012.6233861.

[38] P. Manda, S. Ozkan, H. Wang, F. McCarthy, and S. M. Bridges, "Cross-ontology multi-level association rule mining in the gene ontology," *PLoS ONE*, vol. 7, no. 10, e47411, Oct. 2012. DOI: 10.1371/journal.pone.0047411. [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0047411.

[39] M. Žáková and F. Železný, "Exploiting term, predicate, and feature taxonomies in propositionalization and propositional rule learning," in *Proceedings of the 18th European Conference on Machine Learning and the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD '07)*, Warsaw, Poland, 17-21 September 2007, pp. 798–805.

[40] J. M. Aronis, F. J. Provost, and B. G. Buchanan, "Exploiting background knowledge in automated discovery," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, E. Simoudis, J. Han, and U. M. Fayyad, Eds., AAAI Press, 1996, pp. 355–358, ISBN: 1-57735-004-9. [Online]. Available: http://www.aaai.org/Library/KDD/1996/kdd96-066.php.

[41] G. Garriga, A. Ukkonen, and H. Mannila, "Feature selection in taxonomies with applications to paleontology," in *Proceedings of the 11th International Conference on Discovery Science (DS'08)*, Budapest, Hungary: Springer-Verlag, Berlin Heidelberg, Germany, 13-16 October 2008, pp. 112–123.

[42] M. Záková, F. Zelezný, J. A. García-Sedano, C. M. Tissot, N. Lavrač, P. Kremen, and J. Molina, "Relational data mining applied to virtual engineering of product designs," in *Inductive Logic Programming, 16th International Conference, ILP 2006, Santiago de Compostela, Spain, August 24-27, 2006, Revised Selected Papers*, S. Muggleton, R. P. Otero, and A. Tamaddoni-Nezhad, Eds., ser. Lecture Notes in Computer Science, vol. 4455, Springer, 2006, pp. 439–453, ISBN: 978-3-540-73846-6. DOI: 10.1007/978-3-540-73847-3_39. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-73847-3_39.

[43] M. Hilario, P. Nguyen, H. Do, A. Woznica, and A. Kalousis, "Meta-learning in computational intelligence," in, W. D. K. Grabczewski N. Jankowski, Ed. Springer-Verlag, Berlin Heidelberg, Germany, 2011, ch. Ontology-based meta-mining of knowledge discovery workflows, pp. 273–316.

[44] K. Jörg-Uwe, "Learnability of description logic programs," in *Proceedings of the 12th International Conference on Inductive Logic Programming (ILP'02)*, S. Matwin and C. Sammut, Eds., vol. 2583, Sidney, Australia: Springer Berlin, Heidelberg, Germany, 2002, pp. 117–132, ISBN: 978-3-540-00567-4.

[45] J. Lehmann and C. Haase, "Ideal downward refinement in the EL description logic," in *Proceedings of the 19th International Conference on Inductive Logic Programming (ILP'09)*, Leuven, Belgium: Springer-Verlag, Berlin / Heidelberg, Germany, Feb. 2009, pp. 73–87.

[46] A. Lawrynowicz and J. Potoniec, "Fr-ont: An algorithm for frequent concept mining with formal ontologies," in *Foundations of Intelligent Systems - 19th International Symposium (ISMIS'11)*, Warsaw, Poland: Springer-Verlag, Berlin / Heidelberg, Germany, 28-30 June 2011, pp. 428–437.

[47] B. Berendt, A. Hotho, and G. Stumme, "Towards semantic web mining," in *Proceedings of the International Semantic Web Conference (ISWC'02*, Sardinia, Italy: Springer-Verlag, Berlin / Heidelberg, Germany, Sep. 2002, pp. 264–278.

[48] F. A. Lisi and D. Malerba, "Inducing multi-level association rules from multiple relations," *Machine Learning*, vol. 55, M.-S. Hacid, N. Murray, Z. Ras, and S. Tsumoto, Eds., pp. 175–210, 2 2004, ISSN: 0885-6125. [Online]. Available: http://dx.doi.org/10.1023/B:MACH.0000023151.65011.a3.

[49] F. Lisi and F. Esposito, "Mining the semantic web: A logic-based methodology," in *Foundations of Intelligent Systems*, Springer Berlin / Heidelberg, Germany, 2005, pp. 437–440, ISBN: 978-3-540-25878-0. [Online]. Available: http://dx.doi.org/10.1007/11425274_11.

[50] F. A. Lisi and F. Esposito, "Nonmonotonic onto-relational learning," in *Inductive Logic Programming, 19th International Conference, ILP 2009, Leuven, Belgium, July 02-04, 2009. Revised Papers*, L. D. Raedt, Ed., ser. Lecture Notes in Computer Science, vol. 5989, Springer, 2009, pp. 88–95, ISBN: 978-3-642-13839-3. DOI: 10.1007/978-3-642-13840-9_9. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13840-9_9.

[51] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov, "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 43, pp. 15 545–15 550, Oct. 25, 2005, ISSN: 0027-8424. DOI: 10.1073/pnas.0506580102. [Online]. Available: http://dx.doi.org/10.1073/pnas.0506580102.

[52] S. Y. Kim and D. Volsky, "Page: parametric analysis of gene set enrichment," *BMC Bioinformatics*, vol. 6, no. 1, pp. 144–155, 2005, ISSN: 1471-2105. DOI: 10.1186/1471-2105-6-144. [Online]. Available: http://dx.doi.org/10.1186/1471-2105-6-144.

[53] S. Muggleton, "Inverse entailment and Progol," *New Generation Computing, Special issue on Inductive Logic Programming*, vol. 13, no. 3-4, pp. 245–286, 1995.

[54] P. A. Flach and N. Lachiche, "1BC: A first-order Bayesian classifier," in *Inductive Logic Programming, 9th International Workshop, ILP-99, Bled, Slovenia, June 24-27, 1999, Proceedings*, S. Džeroski and P. A. Flach, Eds., ser. Lecture Notes in Computer Science, vol. 1634, Springer, 1999, pp. 92–103, ISBN: 3-540-66109-3.

[55] M. A. Krogel, S. Rawles, F. Železný, P. A. Flach, N. Lavrač, and S. Wrobel, "Comparative evaluation of approaches to propositionalization," in *Inductive Logic Programming: 13th International Conference, ILP 2003, Szeged, Hungary, September 29-October 1, 2003, Proceedings*, T. Horváth, Ed., ser. Lecture Notes in Computer Science, vol. 2835, Springer, 2003, pp. 197–214, ISBN: 3-540-20144-0.

[56] N. Lavrač, S. Džeroski, and M. Grobelnik, "Machine learning — ewsl-91: European working session on learning porto, portugal, march 6–8, 1991 proceedings," Y. Kodratoff, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, ch. Learning nonrecursive definitions of relations with linus, pp. 265–281, ISBN: 978-3-540-46308-5. DOI: 10.1007/BFb0017020. [Online]. Available: http://dx.doi.org/10.1007/BFb0017020.

[57] S. Kramer, B. Pfahringer, and C. Helma, "Stochastic propositionalization of non-determinate background knowledge," in *Inductive Logic Programming, 8th International Workshop, ILP-98, Madison, Wisconsin, USA, July 22-24, 1998, Proceedings*, D. Page, Ed., ser. Lecture Notes in Computer Science, vol. 1446, Springer, 1998, pp. 80–94, ISBN: 3-540-64738-4.

[58] J. Kranjc, V. Podpečan, and N. Lavrač, "Clowdflows: A cloud based scientific workflow platform," in *ECML PKDD 2012. Proceedings (part II) of the Machine Learning and Knowledge Discovery in Databases - European Conference, Bristol, UK, September 24-28, 2012*, ser. Lecture Notes in Computer Science, P. A. Flach, T. D. Bie, and N. Cristianini, Eds., vol. 7524, Springer, 2012, pp. 816–819, ISBN: 978-3-642-33485-6.

[59] P. A. Flach and N. Lachiche, "IBC: A first-order ayesian classifier," in *Inductive Logic Programming, 9th International Workshop, ILP-99, Bled, Slovenia, June 24-27, 1999, Proceedings*, S. Džeroski and P. A. Flach, Eds., ser. Lecture Notes in Computer Science, vol. 1634, Springer, 1999, pp. 92–103, ISBN: 3-540-66109-3. DOI: 10.1007/3-540-48751-4_10. [Online]. Available: http://dx.doi.org/10.1007/3-540-48751-4_10.

[60] N. Lachiche and P. A. Flach, "1bc2: A true first-order ayesian classifier," in *Inductive Logic Programming, 12th International Conference, ILP 2002, Sydney, Australia, July 9-11, 2002. Revised Papers*, S. Matwin and C. Sammut, Eds., ser. Lecture Notes in Computer Science, vol. 2583, Springer, 2002, pp. 133–148, ISBN: 3-540-00567-6. DOI: 10.1007/3-540-36468-4_9. [Online]. Available: http://dx.doi.org/10.1007/3-540-36468-4_9.

[61] P. A. Flach and N. Lachiche, "Confirmation-guided discovery of first-order rules with Tertius," *Machine Learning*, vol. 42, no. 1/2, pp. 61–95, 2001. DOI: 10.1023/A:1007656703224. [Online]. Available: http://dx.doi.org/10.1023/A:1007656703224.

[62] F. Železný and N. Lavrač, "Propositionalization-based relational subgroup discovery with RSD," *Machine Learning*, vol. 62, no. 1-2, pp. 33–63, 2006.

[63] O. Kuželka and F. Železný, "Block-wise construction of tree-like relational features with monotone reducibility and redundancy," *Machine Learning*, vol. 83, no. 2, pp. 163–192, 2011.

[64] M. A. Krogel and S. Wrobel, "Transformation-based learning using multirelational aggregation," in *ILP 2001. Proceedings of the 11th International Conference on Inductive Logic Programming, Strasbourg, France, September 9-11, 2001*, C. Rouveirol and M. Sebag, Eds., ser. Lecture Notes in Computer Science, vol. 2157, Springer, 2001, pp. 142–155, ISBN: 3-540-42538-1.

[65] C. F. Ahmed, N. Lachiche, C. Charnay, S. E. Jelali, and A. Braud, "Flexible propositionalization of continuous attributes in relational data mining," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7698–7709, 2015. DOI: 10.1016/j.eswa.2015.05.053. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2015.05.053.

[66] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, 1 Nov. 2009, ISSN: 1931-0145. DOI: http://doi.acm.org/10.1145/1656274.1656278. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278.

[67] N. Lavrač, B. Kavšek, P. Flach, and L. Todorovski, "Subgroup discovery with CN2-SD," *Journal of Machine Learning Research*, vol. 5, pp. 153–188, 2004.

[68] W. Hämäläinen, "Efficient search for statistically significant dependency rules in binary data," PhD thesis, Department of Computer Science, University of Helsinki, Finland, 2010.

[69] R. A. Fisher, "On the interpretation of χ^2 from contingency tables, and the calculation of p," English, *Journal of the Royal Stat. Society*, vol. 85, no. 1, pp. 87–94, Jan. 1922, ISSN: 09528385.

[70] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.

[71] Y. Benjamini and D. Yekutieli, "The control of the false discovery rate in multiple testing under dependency," *Annals of Statistics*, vol. 29, no. 4, pp. 1165–1188, 2001.

[72] J. Demšar, "Statistical comparison of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[73] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010, Special Issue on Intelligent Distributed Information Systems, ISSN: 0020-0255. DOI: http://dx.doi.org/10.1016/j.ins.2009.12.010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025509005404.

[74] V. Podpečan, M. Zemenova, and N. Lavrač, "Orange4WS environment for service-oriented data mining," *Computer Journal*, vol. 55, no. 1, pp. 82–98, 2012. DOI: 10.1093/comjnl/bxr077. [Online]. Available: http://dx.doi.org/10.1093/comjnl/bxr077.

[75] C. Sotiriou, P. Wirapati, S. Loi, A. Harris, S. Fox, J. Smeds, H. Nordgren, P. Farmer, V. Praz, B. Haibe-Kains, C. Desmedt, D. Larsimont, F. Cardoso, H. Peterse, D. Nuyten, M. Buyse, M. J. Van de Vijver, J. Bergh, M. Piccart, and M. Delorenzi, "Gene expression profiling in breast cancer: understanding the molecular basis of histologic grade to improve prognosis.," vol. 98, no. 4, pp. 262–272, 2006.

[76] P. Ristoski and H. Paulheim, "Feature selection in hierarchical feature spaces," in *Discovery Science*, ser. Lecture Notes in Computer Science, S. Džeroski, P. Panov, D. Kocev, and L. Todorovski, Eds., vol. 8777, Springer International Publishing, 2014, pp. 288–300, ISBN: 978-3-319-11811-6.

[77] J. Hull, M. Predescu, A. White, J. L. Rotman, J. Fons, L. Gagnon, J. Hyman, H. Hao, L. Johnson, and C. Mann, "The relationship between credit default swap spreads, bond yields, and credit rating announcements," *Journal of Banking and Finance*, vol. 28, pp. 2789–2811, 2004.

Bibliography

Publications Related to the Thesis

Journal Articles

A. Vavpetič and N. Lavrač, "Semantic subgroup discovery systems and workflows in the SDM-toolkit," *The Computer Journal*, vol. 56, no. 3, pp. 304–320, 2013 (included in Chapter 4 of this thesis).

A. Vavpetič, V. Podpečan, and N. Lavrač, "Semantic subgroup explanations," *J. Intell. Inf. Syst.*, vol. 42, no. 2, pp. 233–254, 2014 (included in Chapter 5 of this thesis).

M. Perovšek, A. Vavpetič, J. Kranjc, B. Cestnik, and N. Lavrač, "Wordification: Propositionalization by unfolding relational data into bags of words," *Expert Syst. Appl.*, vol. 42, no. 17-18, pp. 6442–6456, 2015 (not included in this thesis).

P. R. Adhikari, A. Vavpetič, J. Kralj, N. Lavrač, and J. Hollmén, "Explaining mixture models through semantic pattern mining and banded matrix visualization," *Machine Learning Journal*, in press 2016 (included in Chapter 5 of this thesis).

Conference Papers

N. Lavrač, A. Vavpetič, L. Soldatova, I. Trajkovski, and P. Kralj Novak, "Using ontologies in semantic data mining with SEGS and g-SEGS," in *Proceedings of the International Conference on Discovery Science (DS '11)*, Springer, 2011, pp. 165–178.

M. Perovšek, A. Vavpetič, and N. Lavrač, "A wordification approach to relational data mining: Early results," in *Late Breaking Papers of the 22nd International Conference on Inductive Logic Programming, Dubrovnik, Croatia, September 17-19, 2012*, F. Riguzzi and F. Zelezný, Eds., ser. CEUR Workshop Proceedings, vol. 975, CEUR-WS.org, 2012, pp. 56–61. [Online]. Available: `http://ceur-ws.org/Vol-975`.

A. Vavpetič, V. Podpečan, S. Meganck, and N. Lavrač, "Explaining subgroups through ontologies," in *PRICAI 2012: Trends in Artificial Intelligence - 12th Pacific Rim International Conference on Artificial Intelligence, Kuching, Malaysia, September 3-7, 2012. Proceedings*, P. Anthony, M. Ishizuka, and D. Lukose, Eds., ser. Lecture Notes in Computer Science, vol. 7458, Springer, 2012, pp. 625–636, ISBN: 978-3-642-32694-3. DOI: `10.1007/978-3-642-32695-0`. [Online]. Available: `http://dx.doi.org/10.1007/978-3-642-32695-0`.

M. Perovšek, A. Vavpetič, B. Cestnik, and N. Lavrač, "A wordification approach to relational data mining," in *Discovery Science - 16th International Conference, DS 2013, Singapore, October 6-9, 2013. Proceedings*, J. Fürnkranz, E. Hüllermeier, and T. Higuchi, Eds., ser. Lecture Notes in Computer Science, vol. 8140, Springer, 2013, pp. 141–154, ISBN: 978-3-642-40896-0. DOI: `10.1007/978-3-642-40897-7`. [Online]. Available: `http://dx.doi.org/10.1007/978-3-642-40897-7`.

A. Vavpetič, P. K. Novak, and N. Lavrač, "Analysing financial vocabulary using a new semantic subgroup discovery system hedwig," in *Proceedings of the 5th Jožef Stefan International Postgraduate School Students Conference*, Ljubljana, Slovenia, 23 May 2013, pp. 219–229.

A. Vavpetič, P. K. Novak, M. Grčar, I. Mozetič, and N. Lavrač, "Semantic data mining of financial news articles," in *Proceedings of Sixteenth International Conference on Discovery Science (DS 2013)*, J. Fürnkranz, E. Hüllermeier, and T. Higuchi, Eds., ser. Lecture Notes in Computer Science, vol. 8140, Springer Berlin Heidelberg, 2013, pp. 294–307, ISBN: 978-3-642-40896-0 (included in Chapter 5 of this thesis).

P. R. Adhikari, A. Vavpetič, J. Kralj, N. Lavrač, and J. Hollmén, "Explaining mixture models through semantic pattern mining and banded matrix visualization," in *Discovery Science - 17th International Conference, DS 2014, Bled, Slovenia, October 8-10, 2014. Proceedings*, S. Džeroski, P. Panov, D. Kocev, and L. Todorovski, Eds., ser. Lecture Notes in Computer Science, vol. 8777, Springer, 2014, pp. 1–12, ISBN: 978-3-319-11811-6. DOI: 10.1007/978-3-319-11812-3. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-11812-3.

N. Lavrač, M. Perovšek, and A. Vavpetič, "Propositionalization online," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part III*, T. Calders, F. Esposito, E. Hüllermeier, and R. Meo, Eds., ser. Lecture Notes in Computer Science, vol. 8726, Springer, 2014 (included in Chapter 3 of this thesis), pp. 456–459, ISBN: 978-3-662-44844-1. DOI: 10.1007/978-3-662-44845-8. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-44845-8.

N. Lavrač and A. Vavpetič, "Relational and semantic data mining - invited talk," in *Logic Programming and Nonmonotonic Reasoning - 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings*, F. Calimeri, G. Ianni, and M. Truszczynski, Eds., ser. Lecture Notes in Computer Science, vol. 9345, Springer, 2015, pp. 20–31, ISBN: 978-3-319-23263-8. DOI: 10.1007/978-3-319-23264-5. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-23264-5.

Other Publications

Conference Papers

S. Pollak, N. Trdin, A. Vavpetič, and T. Erjavec, "NLP web services for Slovene and English: Morphosyntactic tagging, lemmatisation and definition extraction," *Informatica (Slovenia)*, vol. 36, no. 4, pp. 441–449, 2012.

S. Pollak, A. Vavpetič, J. Kranjc, N. Lavrač, and S. Vintar, "NLP workflow for on-line definition extraction from English and Slovene text corpora," in *11th Conference on Natural Language Processing, KONVENS 2012, Empirical Methods in Natural Language Processing, Vienna, Austria, September 19-21, 2012*, J. Jancsary, Ed., ser. Scientific series of the ÖGAI, vol. 5, ÖGAI, Wien, Österreich, 2012, pp. 53–60, ISBN: 3-85027-005-X. [Online]. Available: http://www.oegai.at/konvens2012/proceedings.shtml.

Biography

Anže Vavpetič was born on June 17, 1987 in Kranj, Slovenia. He finished primary school in Moravče and his secondary education in Ljubljana. In 2006 he started his studies at the Faculty of Computer and Information Science, University of Ljubljana, Slovenia. In 2011 he successfully defended his BSc thesis entitled "The Use of Ontologies as Background Knowledge in Data Mining" under the supervision of Prof. Dr. Igor Kononenko and co-supervision of Prof. Dr. Nada Lavrač.

In 2011 he was accepted to the position of a junior researcher at the Jožef Stefan Institute, Slovenia, under the supervision of Prof. Dr. Nada Lavrač at the Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia. In the same year he started his graduate studies at the Jožef Stefan International Postgraduate School. He enrolled in the PhD programme "Information and Communication Technologies" under the supervision of Prof. Dr. Nada Lavrač.

His research is in the field of data mining and focuses mainly on inductive logic programming, relational data mining and subgroup discovery. More specifically, his research focuses on developing new techniques in automatically exploiting ontological domain knowledge together with multi-relational data in subgroup discovery. He attended several courses and summer schools: on bioinformatics (WSMBio 2012), medical informatics (Systems Medicine of Multifactorial Disorders Workshop & Tutorial) as well as Logic, Language and Information (ESSLLI 2011) and constraint solving techniques (ACAI 2015).

He collaborated in the EU funded FP7 projects e-LICO (An e-Laboratory for Interdisciplinary Collaborative Research in Data Mining and Data-Intensive Science; grant number 231519), ConCreTe (Concept Creation Technology; grant number 611733), and the European Comission Future and Emerging Technologies Flagship project HBP (The Human Brain Project; grant number 604102).