CONSTRUCTION AND EXPLORATION OF REDESCRIPTION SETS

Matej Mihelčić

Doctoral Dissertation
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Supervisor: Asst. Prof. Dr. Tomislav Šmuc, Ruđer Bošković Institute, Zagreb, Croatia
Co-Supervisor: Prof. Dr. Nada Lavrač, Jožef Stefan Institute, Ljubljana, Slovenia, Jožef Stefan International Postgraduate School, Ljubljana, Slovenia

Evaluation Board:
Prof. Dr. Sašo Džeroski, Chair, Jožef Stefan Institute, Ljubljana, Slovenia, Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
Prof. Dr. Marko Robnik Šikonja, Member, Faculty of Computer and Information Science, Ljubljana, Slovenia, Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
Prof. Dr. Hannu Toivonen, Member, University of Helsinki, Helsinki, Finland

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL

Matej Mihelčić

CONSTRUCTION AND EXPLORATION OF REDESCRIPTION SETS

Doctoral Dissertation

GRADNJA IN PREISKOVANJE MNOŽIC POOPISOV

Doktorska disertacija

Supervisor: Asst. Prof. Dr. Tomislav Šmuc

Co-Supervisor: Prof. Dr. Nada Lavrač

Ljubljana, Slovenia, January 2018

Acknowledgments

Reflecting on my education from elementary to postgraduate school, I realized that a large number of people offered their knowledge, experience and wisdom to construct foundations that ultimately enabled the creation of this thesis. Here, I must point out Prof. Dr. Mladen Vuković, whose excellent lectures and patience during the creation of my Master's Thesis motivated this scientific journey. You have my gratitude.

The research period that resulted in the making of this dissertation was a very interesting time for me, a perfect mix of frustration, joy, happiness and excitement. Sincere thanks to my supervisors Asst. Prof. Dr. Tomislav Šmuc and Prof. Dr. Nada Lavrač for making it possible and significantly assisting in making it a successful endeavour. Thank you for your discussions, corrections, guidance and for securing the financial support needed for the completion of this research. I am deeply thankful to Prof. Dr. Nada Lavrač for suggesting redescription mining as a topic of my research and to Asst. Prof. Dr. Tomislav Šmuc for finding sometimes a significant amount of time for discussions about different ideas and potential research directions.

I am very grateful to Prof. Dr. Sašo Džeroski for collaborating with us on this research. Despite his rather busy schedule, he always managed to offer his advice, knowledge and wisdom. Dr. Dragan Gamberger introduced me to the world of multi-view clustering, working with him was a proper introduction to redescription mining. Thank you!

I am grateful for the support received by the following projects: the MAESTRA project (Gr. no. 612944), the MULTIPLEX project (Gr. no. 317532), the Croatian Science Foundation (Pr. no. 9623: Machine Learning Algorithms for Insightful Analysis of Complex Data Structures) and the Knowledge Technologies Research Programme (P2-0103).

Many thanks to Rajko Horvat for maintaining our servers and making prompt interventions when needed. I would like to thank Dr. Esther Galbrun for providing the code of ReReMi algorithm and the Siren tool. Thank you for showing interest in my work. Thanks to Dr. Tomislav Lipić and Dr. Anže Vavpetič for their help with the ClowdFlows technology and to Matija Piškorec for useful discussions about AngularJS, Node.js technologies and giving helpful aesthetics suggestions regarding my figures.

Thanks to Matija for being such a great officemate and to Maria for being great first neighbour. Working and hanging out with you cannot be other than fun. Thanks to my friend Ante for listening about all the status updates throughout these years, during our hikes to Sljeme. Thanks to Dr. Vedrana Vidulin and Dr. Tomislav Lipić for occasional (sometimes late-night) chats about our work, you managed to put my mind at ease and offer useful advice. Thanks to all of my friends at the Ruđer Bošković Institute, especially our little "lunch group". When I was around you, problems seemed solvable and less severe.

Finally, huge thanks go to my family: father Ninel, mother Ankica and sister Natalija. None of this would be possible without your endless love and unlimited support! You were always there for me, for better or worse. Puno vam hvala na svemu! Tata, mama, sestro, baka i dida, puno vas voli vaš Matej.

Abstract

Research in many scientific fields such as physics, biology, medicine, economy etc. requires processing, analysis and understanding of various types of data with the aim of discovering new or analysing existing research hypotheses. Increase of available data and different data sources allows for examining some set of interesting entities from different aspects or using information about different subunits constituting these entities. This enriches the analysis and leads to the discovery of potentially interesting, previously unknown, associations.

Redescription mining, a field of knowledge discovery studied in this thesis, uses data obtained from different sources or describing different aspects of entities, logically organised into disjoint sets of attributes, to re-describe (find different characterizations of) various subsets of given entities. This unsupervised data analysis task is very interesting since it aims to output understandable rule-like descriptions (called redescriptions) that are easy to interpret by the domain experts. By examining the produced redescriptions, researchers can notice interesting connections between the attributes, grouped in different logically connected sets, and can isolate subsets of entities that are interesting for future research.

In the field of biology, it may be interesting to describe different genes based on their occurrence in different life forms, gene functional profile or their pairwise similarity. In the field of economy, more specifically world trade, one can describe different world countries by using information about their trading patterns and their socio-demographic properties. In medicine, relating different types of indicators (such as clinical, biological etc.) may help in explaining the observed symptoms and potentially provide interesting research directions that can lead to the more appropriate patient treatment.

Previous work in redescription mining includes a number of different algorithms that aim to create accurate and statistically significant redescriptions. A tool for redescription exploration and analysis of redescriptions has also been developed with the aim of increasing the overall understanding of produced redescriptions. However, all these techniques focused on producing or analysing individual redescriptions.

This thesis presents developed methods, techniques and tools for redescription mining, redescription set construction, optimization and exploration that use the information about redescriptions and their relations to produce optimized, diverse and accurate sets of redescriptions. These new techniques allow the users to guide redescription set construction to affect the structure of the produced redescription set, enable the use of ensembles of redescription mining methods and provide different modes of targeted and contextual redescription set exploration. The developed techniques have been applied to problems in economy (country trade), medicine (Alzheimer's disease), biology (bioclimatic niches) and social science (co-authorship of scientific papers).

Povzetek

Raziskovanje v mnogih raziskovalnih vedah, kot so fizika, biologija, medicina in ekonomija, zahteva obravnavo, analizo in razumevanje različnih tipov podatkov, s ciljem odkritja novih ali analize obstoječih raziskovalnih hipotez. Naraščajoče količine razpoložljivih podatkov ter različni viri podatkov omogočajo preiskovanje entitet, ki so zanimive z različnih vidikov, kar lahko vodi do odkritja potencialno zanimivih, doslej neznanih povezav.

Rudarjenje poopisov je področje odkrivanja znanja, s katerim se ukvarjamo v pričujoči disertaciji. Pri rudarjenju poopisov uporabljamo podatke, pridobljene z različnih virov, ali podatke, ki opisujejo različne vidike entitet, organizirane v logično nepovezane množice značilk, za poopisovanje (ugotavljanje različnih značilnosti) različnih podmnožic obravnavanih entitet. Ta način nenadzorovane analize podatkov je zanimiv, saj generira razumljive, pravilom-podobne opise (imenovane poopisi), ki so za domenske strokovnjake lahko razumljivi. S preiskovanjem generiranih poopisov lahko raziskovalci ugotovijo zanimive povezave med značilkami, grupiranimi v različne logično povezane množice značilk, kar omogoča odkrivanje podmnožic entitet, zanimivih za nadaljnje raziskovanje.

Na področju biologije je zanimivo opisati različne gene na podlagi njihovih pojavitev v različnih življenjskih oblikah, profilih genskih funkcij ali njihovih medsebojnih podobnostih. Na področju ekonomije je npr. na področju svetovne trgovine mogoče opisati različne države sveta z informacijami o vzorcih trgovanja in socio-demografskimi značilnostmi držav. V medicini lahko povezovanje različnih kazalcev (kot so klinični, biološki itd.) pomaga pojasniti ugotovljene simptome ter potencialno usmeriti raziskovanje na nova področja, s katerimi bi lahko zagotovili ustrezno zdravljenje.

Prejšnje raziskave na področju rudarjenja poopisov vključujejo razvoj različnih algoritmov, ki poskušajo generirati natančne in statistično pomembne poopise. Orodja za preiskovanje in analizo poopisov so prav tako bila razvita s ciljem boljšega splošnega razumevanja generiranih poopisov. Vendar so bile vse doslej razvite metode osredotočene na izdelavo ali analizo posamičnih poopisov.

V pričujoči disertaciji predstavimo razvite metode, tehnike in orodja za rudarjenje poopisov, gradnjo množic poopisov ter optimizacijo in analizo poopisov, ki uporabljajo informacije o posameznih poopisih in njihovih odnosih za pridobitev optimiziranih, raznovrstnih in natančnih množic poopisov. Te nove tehnike omogočajo uporabnikom vpliv na strukturo pridobljenih množic poopisov, omogočajo uporabo ansamblov metod za rudarjenje poopisov ter zagotavljajo različne načine usmerjenih in kontekstualno odvisnih analiz množic poopisov. Razvite metode so uporabljene na problemih s področja ekonomije (svetovna trgovina), medicine (Alzheimerjeva bolezen), biologije (bioklimatske niše) ter družbenih ved (soavtorstvo znanstvenih člankov).

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

AD	Alzheimer’s disease
ANG2	Angiopoietin-2
APOAII	Apoliprotein A-II
APOB	Apoliprotein B
BNP	Brain natriuretic peptide
CBRM	Constraint-based redescription mining
CLUS-RM	Redescription mining algorithm using Predictive Clustering trees that are implemented in the CLUS decision tree and rule induction system.
CN	Control normal
CNTF	Ciliary neurotrophic factor
DM	Decision maker
DNF	Disjunctive Normal Form
EMCI	Early mild cognitive impairment
FASL	Fas ligand
INSULIN	Insulin
IPS	International Postgraduate School
ISCA	International Symposium on Computer Architecture
JSI	Jožef Stefan Institute
LEPTIN	Leptin
LMCI	Late mild cognitive impairment
MCDA	Multi-criteria decision aid
MCRPHMIF	Macrophage migration inhibitory factor
PAPP-A	Pregnancy-associated plasma protein A
PCT	Predictive Clustering trees
PPP	Pancreatic polypeptide
RM	Redescription mining
SMC	Significant memory concern
SPARE_AD	Spatial Pattern of Abnormalities for Recognition of Early AD
TSTSTRNT	Total blood testosterone

Chapter 1

Introduction

Knowledge discovery in databases is the process of identifying valid, novel, potentially useful, and ultimately understandable knowledge in data [1]. This process encompasses data set construction, knowledge extraction, exploration, analysis, presentation and maintenance of discovered knowledge. The process of identification and extraction of knowledge from data is the main goal of a field of computer science called *data mining* [2]. Depending on the type of wanted knowledge and the structure of available data, different fields of data mining with a large variety of accompanying methods have been developed to assist the identification and extraction of useful knowledge from data. Methods and algorithms for data mining often utilize different concepts from mathematics and other fields of computer science, such as artificial intelligence and machine learning, to enhance the process of discovering interesting patterns.

Our research is directed to the task called *redescription mining*. Redescription mining [3] is a field of data mining that aims to find different descriptions (characterizations) of the same or very similar subsets of entities or to discover subsets of entities that can be characterized in multiple ways. Such sets of entities, originally described by different sets of descriptors, allow explaining the underlying problem from different perspectives which may lead to better understanding of the data or discovery of new research hypotheses. The output of the redescription mining process is a list of multiple descriptions (tuples of logical formulas) which are called *redescriptions*. Current approaches for redescription mining output redescriptions containing only pairs of descriptions.

In this thesis we tackle two very important tasks that emerge within redescription mining: *redescription set construction* and *redescription set exploration*. Redescription set construction consists of discovery, selection, filtering and arranging of discovered redescriptions into a set that is presented to the user. This is naturally complemented by the task of redescription set exploration, which aims to provide methods and techniques for in-depth exploration of constructed sets of redescriptions.

1.1 Motivation

Data analysis has become a very important technique to obtain new knowledge or increase understanding of the underlying problem. Because of this, it is omnipresent in various scientific fields, but also increasingly used in many industrial applications. Data often contains a large number of different attributes, obtained from different sources, produced by different experiments— representing different aspects of the observed entities. Understanding such data, especially the interactions between different attributes, provides insights that may lead to new scientific discoveries or e.g., to knowledge that can be used to increase the profit of some company.

Redescription mining provides means to describe different subsets of entities in the data and reveal bi-directional associations between various attributes from different views or contexts. For instance, it may be interesting for economists to analyse the world countries by using socio-demographic characteristics and their trading patterns [4]–[6]. Take an example from biology, where the interested user would like to describe different geographical locations by using information about the weather and information about the habitat of different animal species [7]–[9]. In this application, redescription R_{ex} defined in Example 1.1 re-describes 35 different locations in Europe by using information about species inhabiting these locations, in our example the Polar bear, and several climate conditions at these locations such as maximum temperature in November t_{11}^{+} and minimal temperature in July t_{7}^{-}.

Example 1.1. $R_{ex} = (q_{1_{ex}}, q_{2_{ex}})$, where $q_{1_{ex}} =$ (Polar Bear = TRUE) and $q_{2_{ex}} = -7.9 \leq t_{11}^{+} \leq -4.1 \ \wedge \ -2.0 \leq t_{7}^{-} \leq 2.7$

Redescription mining also allows relating a set of well-known or fully explored attributes with a set of unexplored attributes describing different properties of an entity. In such tasks, redescription mining can help in understanding the functions and interactions between the mostly unexplored attributes. For example, patients may be subject to a number of different clinical tests and diagnoses targeted at testing different body functions. The results of such tests can be interpretable and well-understood. For instance, low score on different memory or cognitive tests can indicate abnormalities in different parts of human brain. However, understanding the detected levels of many different biological indicators, such as levels of different hormones, neural activity, abnormalities in different genetic or blood test markers and the effects of such tests on the overall human health may not be well-understood. Relating clinical with biological attributes may assist in understanding the role and function of different biological properties on the human health or may indicate the targets for treating various diseases.

Several algorithms exist that use different techniques to find redescriptions [3], [10]–[14], however they consider redescriptions in isolation and aim to produce sets of individual, highly accurate redescriptions. Similarly, an existing tool for redescription exploration [15] provides different methods for the analysis of individual redescriptions. However, none of the existing tools use information about previously produced redescriptions and redescription set properties during redescription construction or exploration.

This thesis presents methods and techniques aimed to create sets of redescriptions with some desired properties. Instead of producing redescription sets that contain highly accurate redescriptions, the aim is to construct redescription sets in such a way to increase the favourable redescription and redescription set properties. By producing *optimized redescription sets*, the users will be able to have deeper insight and a larger influence on the properties of the resulting redescription sets, with the goal of obtaining useful knowledge. Similarly, the information derived from the redescription sets is used to enhance different modes of in-depth redescription set exploration.

1.2 Hypotheses

The main hypothesis explored in this thesis is that using information about redescription sets, instead of observing individual redescriptions in isolation, allows for the construction of redescription sets with superior properties in terms of redescription accuracy, diversity, complexity and other potentially interesting properties. Furthermore, it allows for using all the produced redescriptions in order to improve the properties of redescription sets under construction and provides more flexibility and control over the structure of the final redescription sets by using user-defined preferences. In the context of redescription set optimization, the hypothesis is that a larger number of highly accurate, diverse redescriptions increases the effectiveness of redescription set optimization. In addition, we hypothesize that a high number of accurate redescriptions may be produced by using multi-target regression Predictive Clustering trees [16], [17] algorithm to guide the search in redescription mining process. Finally, using information about redescription sets provides the means to perform more structured, context-driven redescription set exploration.

1.3 Objectives

The main objective of this thesis is to develop new redescription mining methods and techniques for insightful data analysis. To reach this objective, we need to solve several related problems. The first step involves obtaining a diverse set of redescriptions from the available data. Due to the potentially large size of the obtained knowledge base, it is necessary to develop efficient filtering and selection techniques that enable the selection of a smaller number of interesting patterns that are presented to the user. Such smaller sets are easier to process and understand. This leads to our second task of developing novel techniques to reduce the amount of knowledge returned to the user but keeping the essentials in the predefined boundaries. The idea is to reduce the amount of returned redescriptions but also allow influencing and structuring the generated redescription sets by using different user-defined criteria. The final step, that brings us closer, to achieving our goal of insightful data analysis with redescription mining is to provide methods, techniques and tools for redescription set exploration. These techniques are devised to help efficient exploration, analysis, filtering and selection of redescriptions with the goal of obtaining valuable new insights and information from the produced redescription sets. In all previously described steps, the emphases is on the importance and use of the information about the structure of the obtained knowledge base.

1.4 Contributions

In this thesis we present the following original contributions:

1. Redescription mining algorithm CLUS-RM that uses multi-target regression and classification Predictive Clustering trees (PCT) [16], [17] as means for redescription creation. This allows utilizing multi-label classification and multi-target regression capabilities of PCT for redescription construction which increases the accuracy, diversity and number of produced redescriptions. The algorithm is extended with a random forest of Predictive Clustering Trees to create the building blocks for redescriptions (publication [18]). This extension further increases the accuracy and diversity of produced redescriptions. The algorithm is described in Sections 4.3 and 4.4.

2. Two methods for redescription set optimization that select and arrange redescriptions in a redescription set to maximize a set of predefined criteria:

   a) *Optimization by redescription exchange*, which enables using redescription mining algorithms without specifying the minimal redescription accuracy constraint—usually determined through experimentation. This optimization approach is presented in publications [18], [19].

   b) *Optimization by redescription extraction* (presented in publication [20]) effectively allows the use of ensembles of redescription mining algorithms and enables creating multiple redescription sets with different properties by using only one execution of a redescription mining algorithm.

Redescription set optimization procedures are described in Sections 5.2, 5.3 and 5.4.

3. A methodology and accompanying tool for redescription set exploration (presented in publication [21]) that uses various information derived from the redescription set and information about individual redescriptions to provide new insights and make the exploration of redescription sets more efficient. This contribution is described in Section 6.3.

4. An algorithm for *Constraint-based redescription mining* that allows several modes of exploration using user-defined attribute constraints. It is noted that the entity constraints can be introduced analogously to the attribute constraints. The proposed extensions are presented in publication [22] and described in Section 4.5.

5. Two new redescription evaluation measures have been constructed (*query non-missing Jaccard index* and *variability index*) and several evaluation measures have been adopted and used in the context of redescription mining (*redescription entity redundancy*, *redescription attribute redundancy*, *redescription complexity* and *redescription coverage*). All these measures can be used to evaluate the properties of produced redescription sets. The aforementioned measures are presented in publication [19], [20] and described in Section 3.2.

6. A technique for redescription query size reduction (presented in publication [19]) and a technique for iterative increase of redescription accuracy called *conjunctive refinement* (presented in publication [20]) are described in Section 4.6.

7. The developed methodology was successfully applied on several use-cases from medicine (on the Alzheimer's disease ADNI dataset [23], publication [22]), economy (on the Country dataset [4]–[6], publications [18]–[21]), biology (on the Bio dataset [7]–[9], publications [18], [20]) and in social network analysis (on the DBLP dataset [9], [24], publications [18], [20]).

The main contributions are demonstrated in Figure 1.1.

1.5 Main Publications Related to the Thesis

Journal articles

[18] M. Mihelčić, S. Džeroski, N. Lavrač, and T. Šmuc, "Redescription mining augmented with random forest of multi-target predictive clustering trees," *Journal of Intelligent Information Systems*, pp. 1–34, 2017, In press.

[20] M. Mihelčić, S. Džeroski, N. Lavrač, and T. Šmuc, "A framework for redescription set construction," *Expert Systems with Applications*, vol. 68, pp. 196–215, 2017, ISSN: 0957-4174.

[22] M. Mihelčić, G. Šimić, M. Babić Leko, N. Lavrač, S. Džeroski, T. Šmuc, and for the Alzheimer's Disease Neuroimaging Initiative, "Using redescription mining to relate clinical and biological characteristics of cognitively impaired and alzheimer's disease patients," *PLOS ONE*, vol. 12, no. 10, pp. 1–35, 2017.

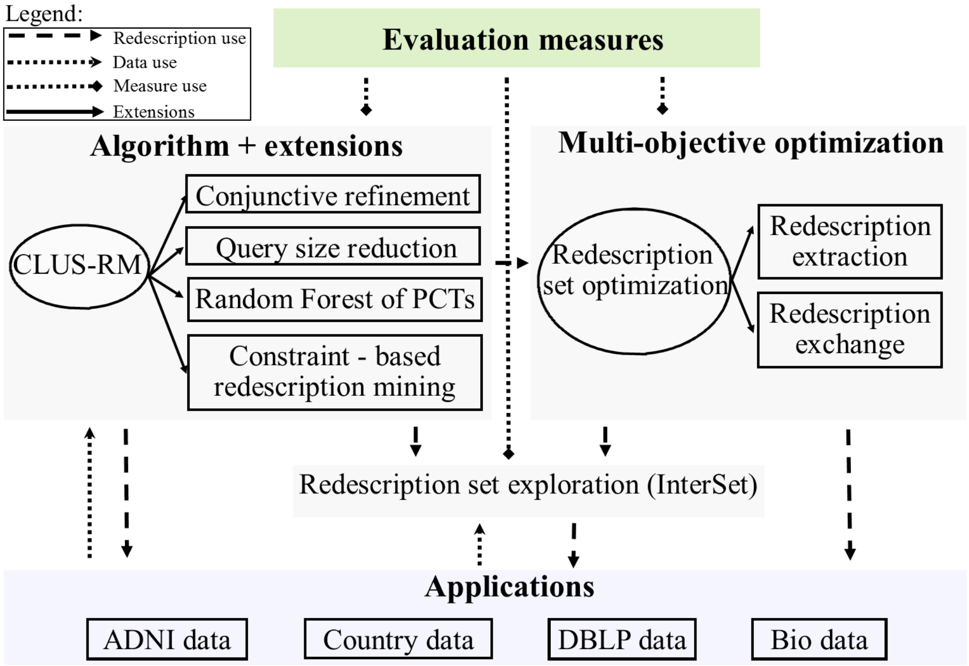


Figure 1.1: The main contributions presented in this thesis.

Conference and workshop papers

[19] M. Mihelčić, S. Džeroski, N. Lavrač, and T. Šmuc, "Redescription mining with multi-target predictive clustering trees," in *Proceedings of the 4th International Workshop, New Frontiers in Mining Complex Patterns, NFMCP 2015, Held in conjunction with ECML-PKDD 2015, Porto, Portugal, September 7, 2015, Revised Selected Papers*, M. Ceci, C. Loglisci, G. Manco, E. Masciari, and Z. W. Ras, Eds. Cham: Springer International Publishing, 2016, pp. 125–143.

[21] M. Mihelčić and T. Šmuc, "InterSet: Interactive redescription set exploration," in *Proceedings of Discovery Science: 19th International Conference, DS 2016, Bari, Italy, October 19–21, 2016*, T. Calders, M. Ceci, and D. Malerba, Eds. Cham: Springer International Publishing, 2016, pp. 35–50.

1.6 Thesis Structure

The rest of this thesis is structured as follows. Chapter 2 provides background information and corresponding notation for redescription mining and related areas. Chapter 3 provides the necessary definitions and analyses various redescription evaluation measures. Chapter 4 introduces Predictive Clustering Trees, related redescription mining approaches and describes the CLUS-RM algorithm and its extensions (the first major contribution of the thesis). Chapter 5 introduces the multi-objective optimization and explains the second contribution: redescription set optimization. Chapter 6 discusses interactive data mining techniques and focuses on interactive redescription mining and exploration. This chapter also provides information about the third contribution: the approach for interactive redescription set exploration realized through the tool InterSet. Chapter 7 explains the results of evaluation, while Chapter 8 discusses the application of redescription mining in the field of medicine (redescribing subjects suffering from a different level of cognitive impairment or Alzheimer's disease). Chapter 9 discusses software availability while Chapter 10 presents the overall conclusions based on the contributions presented in this thesis. Thesis contains two Appendices: Appendix A demonstrates correspondence between redescription set and multi-objective optimization while Appendix B analyses CLUS-RM algorithm and the produced redescriptions with respect to predictivity and generalizability. Statistical significance of produced redescriptions is further evaluated using permutation tests and corrections for multiple hypothesis testing.

Chapter 2

Background

Redescription mining is a descriptive, unsupervised knowledge discovery task aimed at finding redescriptions of different subsets of entities by using one or more disjoint sets of attributes (called *views*). In general, entities can be redescribed by using data from different sources, describing different contexts of entities or obtained from different experiments.

To explain the input and output of the redescription mining task, we provide a practical example involving a set of geographical locations in Europe.

Table 2.1: Input example for the Bio dataset.

(a) View 1: Table containing information about habitation of different mammalian species.

Locations	**Crete Spiny Mouse**	**Moose**	...	**Marbled Polecat**	**Red Fox**
$26SLH1$	false	false	...	false	false
.	.	.	...	.	.
.	.	.	...	.	.
.	.	.	...	.	.
$30VVJ3$	false	false	...	false	true
.	.	.	...	.	.
.	.	.	...	.	.
.	.	.	...	.	.
$34UFB3$	false	true	...	false	true
.	.	.	...	.	.
.	.	.	...	.	.
.	.	.	...	.	.
$36WVD2$	false	true	...	false	true

(b) View 2: Table containing information about weather conditions (monthly maximum, minimum, average temperatures and average precipitations).

Locations	t_1^-	...	t_{12}^-	t_1^+	...	t_{12}^+	$\widetilde{t_1}$	...	$\widetilde{t_{12}}$	$\widetilde{p_1}$	...	$\widetilde{p_{12}}$
$26SLH1$	10.6	...	11.6	14.8	...	15.8	12.7	...	13.7	130.0	...	126.0
.	.	...	.	.	...	.	.	...	.	.	...	.
.	.	...	.	.	...	.	.	...	.	.	...	.
.	.	...	.	.	...	.	.	...	.	.	...	.
$30VVJ3$	−3.0	...	−1.8	5.6	...	6.4	1.32	...	2.27	74.0	...	74.6
.	.	...	.	.	...	.	.	...	.	.	...	.
.	.	...	.	.	...	.	.	...	.	.	...	.
.	.	...	.	.	...	.	.	...	.	.	...	.
$34UFB3$	−8.1	...	−5.1	−1.7	...	0.2	−4.88	...	−2.44	22.44	...	31.4
.	.	...	.	.	...	.	.	...	.	.	...	.
.	.	...	.	.	...	.	.	...	.	.	...	.
.	.	...	.	.	...	.	.	...	.	.	...	.
$36WVD2$	−9.7	...	−8.4	−2.0	...	−0.7	−5.64	...	−4.34	50.1	...	45.9

The data [7]–[9] contain a set of attributes describing habitation of different mammalian species at each location, and a set of attributes describing weather conditions at these locations. As a motivating example, we provide the input data shown in Table 2.1, containing two views. Each code in Table 2.1 represents one location in Europe, $26SLH1$ represents the geographical location with Latitude 38.61 and Longitude -29.01 $(38.61, -29.01)$, $30VVJ3$ is a location with coordinates $(57.52, -3.42)$, $34UFB3$ is located at $(51.2, 23.57)$ and $36WVD2$ represents the location with coordinates $(70.52, 30.83)$.

The output of redescription mining is a set of redescriptions (tuples of logical formulas). Redescription R_{ex_1} defined in Example 2.1 describes 50 different locations located on the Iberian Peninsula (the exact locations can be seen in Figure 2.1).

These geographical locations are inhabited by Egyptian Mongoose and Wood mouse. The average precipitation in October on these locations is between 55.1 and 84 mm, in August between 1.6 and 7.4 mm and in May between 27.6 and 55.8 mm.

Example 2.1. $R_{ex_1} = (q_{1_{ex_1}}, q_{2_{ex_1}})$, where $q_{1_{ex_1}} =$ (Egyptian Mongoose = TRUE) $\wedge$ (Wood mouse = TRUE) and $q_{2_{ex_1}} = 55.1 \leq \widetilde{p_{10}} \leq 84.0 \wedge 1.6 \leq \widetilde{p_8} \leq 7.4 \wedge 27.6 \leq \widetilde{p_5} \leq 55.8$

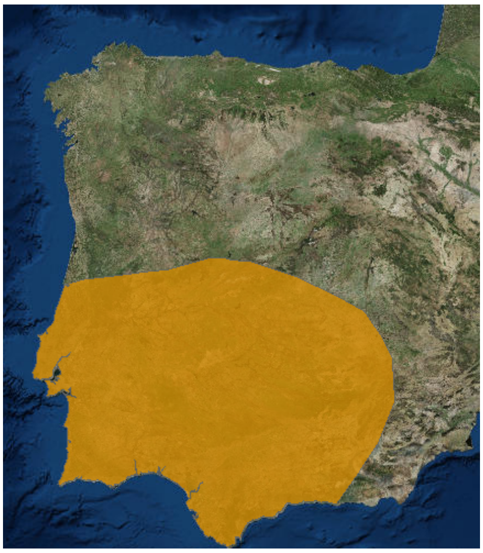


Figure 2.1: Part of Iberian Peninsula described by redescription R_{ex_1}. Geographical map was created with the tool freely available online at `http://maps.iucnredlist.org`. Last access, 07.01.2018.

2.1 Problem Definition

In this section, we formally define redescription mining and introduce the notation that will be used throughout this thesis. It is similar to the notation used in [9].

For a set of entities E and a set of attributes $\mathcal{A}$ which are grouped in a set of views $\mathcal{W}$ (disjoint attribute sets), a function $v : \mathcal{A} \mapsto \mathcal{W}$ maps each attribute to the corresponding view. The input dataset D is defined as $D = (E, \mathcal{A}, v)$. Redescription mining in general considers redescriptions constructed on a set of views $\mathcal{W} = \{W_1, W_2, \dots, W_n\}$, $n \geq 1$, however like in all currently developed redescription mining approaches, we also use maximally two views $\{W_1, W_2\}$. Because of this, all further notation will be restricted to the case of two views. Entity $e_i, i \leq |E|$ has value $W_k(i, j)$ of attribute $a_j, j \leq |V_k|$ contained in view W_k, $k \in \{1, 2\}$. We use V_1, V_2 to denote a set of attributes mapped to views W_1 and W_2.

A logical formula comprised of attributes from $\mathcal{A}$ (with the appropriate conditions), which are connected with some logical operators (conjunction, disjunction and negation are used in redescription mining) is called a *query*. $q1_{ex_1}$ and $q2_{ex_1}$ from Example 2.1 are two queries. A query language Q defines a set of valid queries. Depending on the type of redescription mining task (for instance in relational redescription mining [25]), the language can be extended with some additional operators. The set of entities described by a query q is called its support set[1] and is denoted $supp(q)$. For a given query q, we define $attrs(q)$ as a set of attributes constituting the query, and with $attr(q)$ a multi-set of attributes constituting this query.

Redescription mining provides means to study the relations between queries obtained on different views. A general relation $\sim$ is used to denote the similarity of support sets of two queries[2]. A *redescription* $R = (q_1, q_2)$ is defined as a pair of queries, one for each view in the data and its support set is the set of entities described by both queries that constitute this redescription: $supp(R) = supp(q_1) \cap supp(q_2)$. The support set of redescription R_{ex1} from Example 2.1 is shown in Figure 2.1. We use $attr(R)$ to denote the multiset of attributes used in the redescription R and $attrs(R)$ to denote the corresponding set of attributes. Formally, $attr(R) = (attrs(R), \mathcal{M})$, where $\mathcal{M} : attrs(R) \mapsto \mathbb{N}^+$. In the presented example, $attr(R_{ex1}) = attrs(R_{ex1}) = \{$Egyptian Mongoose, Wood mouse, $\widetilde{p_5}$, $\widetilde{p_8}$, $\widetilde{p_{10}}\}$.

Formally, redescriptions and redescription mining are defined as (see [9]):

Definition 2.2. Given a dataset $(E, \mathcal{A}, v)$, a query language Q over $\mathcal{A}$ and a binary relation $\sim$, a *redescription* is a pair of queries $(q_1, q_2) \in Q \times Q$ such that $attrs(q_1) \cap attrs(q_2) = \emptyset$ and $supp(q_1) \sim supp(q_2)$.

Definition 2.3. Given a dataset $(E, \mathcal{A}, v)$, a query language Q over $\mathcal{A}$, a binary relation $\sim$, and a set of constraints $\mathcal{C}$, *redescription mining* is a task of finding redescriptions that satisfy constraints in $\mathcal{C}$.

A set of constraints $\mathcal{C}$ is defined by the user at the beginning of the redescription mining task to constrain certain redescription properties. The constraints imposed on redescriptions define minimum redescription support set size, minimal redescription accuracy, the significance level (for a detailed explanation and definition of redescription accuracy and statistical significance we refer the reader to Chapter 3).

We use $\mathcal{R}$ to denote a redescription set containing all the constructed redescriptions. $\mathcal{R}_{opt}$ denotes an optimized subset of $\mathcal{R}$ constructed according to user-defined importance preferences on various redescription quality criteria and incorporating the constraint on the size of redescription set $\mathcal{R}_{opt}$.

2.2 Features and Queries

In this section, we describe different types of features and queries used in redescription mining that are related to the work presented in this thesis.

If some input dataset contains only attribute values describing entities (as in the example from Table 2.1), it is called a *propositional* dataset (see [9]). If it contains information about the relations between different entities, possibly in addition to the attribute values, it is a *relational* dataset. In this work, we concentrate exclusively on propositional datasets.

Each attribute contained in the dataset is characterized by its type (Boolean, categorical, numerical) and its range (a set of possible values). Let us suppose that we have one Boolean attribute B, one categorical attribute C with possible values *windy*, *sunny*, *rainy* and *cloudy* and a numerical attribute N with values contained within the $[-5, 5]$ interval.

[1] The term *support* is used inconsistently to denote the support set or the cardinality of support set interchangeably in [9], [13], [18], [19] etc.

[2] E.g., it can be any of the similarity measures defined in Section 3.2.

The range of attribute B is $R_B = \{\text{true}, \text{false}\}$, which can be also written as $R_B = \{1, 0\}$. $R_C = \{\text{windy, sunny, rainy, cloudy}\}$ and $R_N = \{x,\ x \in [-5, 5]\}$. Taking a subset of an attribute range, for instance $R_S \subset R_C$, can be used to define a truth value assignment which assigns a logical value *true* to all entities with a value in R_S and *false* to all other entities.

Placing such constraints on the attribute range, defines constructs called *features* [26] which are used to construct queries.

2.2.1 Features

Depending on the attribute type, the features can be:

- Boolean features—restrict the attribute value either to *true* ($B = \text{true}$) or to *false* ($B = \text{false}$). In case of Boolean features, feature $B = \text{true}$ is often written as B while the proposition $B = \text{false}$ is written as $\neg B$. All instances with the value *true* satisfy the condition imposed by the feature $B = \text{true}$. This is denoted as $[B = \text{true}]$ by using Iverson bracket (see [9]), which evaluates to 1 if the condition $B = \text{true}$ is satisfied and to 0 otherwise.

- Categorical or nominal features—restrict the attribute value to one of the categories contained in the attribute range. For instance, $C = \text{sunny}$ is a categorical feature. All entities with a categorical value *sunny* satisfy the condition imposed by the feature $C = \text{sunny}$. The corresponding truth value assignment is denoted $[C = \text{sunny}]$.

- Real-valued features—restrict the attribute value to a specific interval. For instance, $x \leq N \leq y$, where $x, y \in [-5, 5]$, $x \leq y$ is a numerical feature. All entities with a numerical value contained in the $[x, y]$ interval satisfy the conditions imposed by the feature $x \leq N \leq y$. The corresponding truth value assignment is denoted $[x \leq N \leq y]$.

2.2.2 Queries

Features can be combined with different logical operators (conjunction $\wedge$, disjunction $\vee$ and negation $\neg$) to form queries. The support set of a query is computed by observing the entities that satisfy individual features building this query. We state several claims that are valid if there are no missing values in the dataset.

If two sets of entities S_1 and S_2 satisfy the conditions of features P_1 and P_2 respectfully, a query defined as $q_c = P_1\ \wedge\ P_2$ has a support set equal $S_1 \cap S_2$. Similarly a query defined as $q_d = P_1\ \vee\ P_2$ has a support set $S_1 \cup S_2$. In general, for a query defined as $q = P$, where P is some proposition that defines conditions satisfied by entities contained in set S, a query $q' = \neg\ P$ has a support set $E \setminus S$.

For a Boolean proposition B and a corresponding set of entities S, query $q = \neg\ B$ has a support set equivalent to the set of entities that satisfy the properties of a proposition $q' = (B = \text{false})$. For nominal propositions (e.g. $C = \text{rainy}$), the query defined as $q = \neg\ C = (\neg\ \text{rainy})$ has the same support set as defined by a query $q' = (C = \text{sunny})\ \vee\ (C = \text{windy})\ \vee\ (C = \text{cloudy})$ and in the case of real-valued propositions, query $q = \neg\ (0.2 \leq N \leq 3.3)$ has a support set equivalent to that of a query $q' = N < 0.2\ \vee\ N > 3.3$.

Depending on the form, queries can be divided into:

- *Monotone conjunctions* are the most restricted queries. In such queries, features are combined using only conjunction operators.

- *Unrestricted queries* are queries where features can be combined using any operator with no special limits other than those defined by operator definitions.

- *Linearly parsable queries* used by Galbrun and Miettinen in [13]. These queries are constructed following a generative grammar of the restricted query language designed so that the resulting queries can be evaluated from left to right irrespective of the binary operators precedence (for more details see [9]).

Queries produced within the CLUS-RM redescription mining algorithm (explained in Section 4.3) have the form of monotone conjunctions. However, they can be further refined in redescription construction by applying the disjunction operator in combination with other queries and the negation operator which results in unrestricted queries. A positive property of the resulting queries contained in redescriptions is that they can be easily transformed to the Disjunctive Normal Form (DNF) [27], which is more understandable and enables thorough query analysis.

2.3 Redescription Construction Strategies

Redescriptions can be constructed by using different strategies. These strategies differ in the methodology used to construct redescription queries and the approach of combining these queries into redescriptions. Different redescription mining algorithms, also presented in Section 2.4, can be divided by the underlying construction strategy used in the redescription mining process.

2.3.1 Query mining and pairing

This is the simplest construction strategy that consists of mining redescription queries from the dataset and combining produced queries into redescriptions. Since the methodology does not implement any mechanism of guiding the search to produce compatible queries, it relies on pure chance to obtain redescriptions satisfying predefined quality criteria. If the number of available views is small, mining queries from each view separately and combining them into redescriptions might be feasible. However, even in such cases, many query pairs could produce unsatisfactory redescriptions. When the number of views is large, this kind of approach needs to test many combinations of queries with significantly smaller probability of creating highly accurate redescription. Thus, it might be preferable to combine all views into one large view containing all attributes and use the mining methodology to create queries which are later used to construct redescriptions. A potential problem with this approach is that all produced queries contain attributes from multiple different views, which completely obfuscates the separation of views and makes the analysis significantly harder.

The main advantage of this scheme is that it allows using a different association rule [28]–[30] and different types of itemset mining [31]–[33] algorithms for redescription construction.

The query mining and pairing approach is described in Figure 2.2. Queries are mined from the corresponding views with some rule-producing methodology such as association rule mining, frequent itemset mining etc. Queries are in a form $q = a_{c(1)} \ \wedge \ a_{c(2)} \cdots \wedge a_{c(k)}$. Thus, they contain a different number of attributes. The pairing process consists of computing the Cartesian product of two query sets and filtering with respect to user-defined quality criteria such as Jaccard index for redescription accuracy, p-value, support etc. (for more details see Chapter 3).

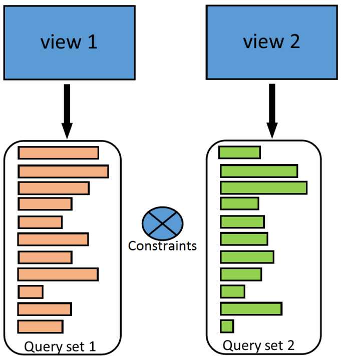


Figure 2.2: Illustration of query mining and pairing approach.

Redescription mining approaches [11], [10] and the MID approach presented in [12] use the query mining and pairing exploration strategy to construct redescriptions.

2.3.2 Greedy atomic updates

The greedy atomic updates exploration strategy is an iterative greedy approach that improves the constructed redescriptions by iteratively adding, removing or modifying attributes contained within redescription queries by using attributes from different views. The main advantage of this approach is that it offers the ability to introduce more granular updates. Performing greedy atomic updates in the alternation mode (adding update by using attributes from one view followed by using attributes from the second) allows guiding query construction at each step, as opposed to the query mining and pairing methodology. The main drawback of this methodology is that the exhaustive search requires testing all attribute combinations. Because of this, methods that use greedy atomic updates usually restrict the number of updates allowed per query.

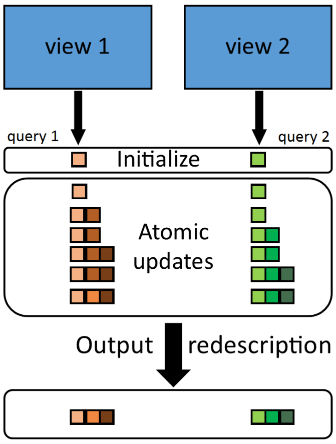


Figure 2.3: Illustration of the greedy atomic updates process.

The greedy atomic updates process is illustrated in Figure 2.3. The initialization process consists of selecting a pair of attributes that are combined into redescriptions. The selection is based on the accuracy of the produced redescription. Each greedy atomic update adds/deletes one attribute or updates the attribute value. The redescription that can not be improved further and that satisfies user-defined constraints is returned to the user. The entire process is repeated for a specified number of iterations with different initial attribute pairs, which allows producing different redescriptions.

This redescription exploration approach, restricted to the addition of features, was used in the greedy approach by Gallo et al. [12]. The approach used the single best redescription at each update step obtained for a given initial pair of attributes. It was extended by Galbrun and Miettinen [13] with the addition of a beam search procedure that saves multiple top candidates at each step instead of focusing only on the best improvement.

2.3.3 Alternating scheme

The alternating scheme requires performing the initialization step in which the first query set is produced. It can be performed in several ways, for instance by randomly splitting the entities and using this split for query construction with the use of classification algorithms (capable of producing rules or transformable to rules) or by using queries containing only one attribute as initialization.

The initially obtained queries are used as a starting point for the alternating scheme, which attempts to find a good matching query to obtain accurate redescriptions. The procedure iterates by replacing redescription queries with newly constructed queries that increase the accuracy of a given redescription. The procedure is repeated until the predefined number of iterations has been reached or no further improvements can be made.

The connection between different queries and the search for a matching query can be achieved with the classification algorithms. The entities described by one query can be considered targets for the classification task from which the second (matching) query is constructed. One of the main goals is to obtain interpretable queries which can be combined into redescriptions, thus tree-based classifiers (such as CART [34] or Predictive Clustering Trees [16], [35], [36] etc.) or rule-based classifiers (such as Predictive Rules [37] etc.) can be used in this process. The alternating scheme is illustrated in Figure 2.4.

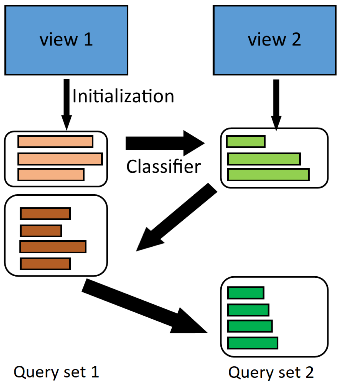


Figure 2.4: Illustration of the alternating scheme for redescription exploration and construction.

The alternating approach is similar to the mining and pairing approach because it uses fully constructed queries to create redescriptions. However, the main and most important difference is that it selects the matching queries in a targeted manner as opposed to the query and pairing approach that tries all possible combinations. Compared to greedy atomic updates, it scales better with respect to the number of attributes but is more sensitive to the number of available entities.

The alternating scheme was introduced by Ramakrishnan et al. [3] and incorporated into the CARTwheels algorithm. It is also used in SplitTrees and LayeredTrees algorithms [14], [38]. Galbrun and Kimming [25] use the alternating scheme for mining relational redescriptions. The algorithm CLUS-RM, described in Chapter 4 and introduced in [18], [19] is based on the alternating scheme, however it also incorporates components from the greedy atomic updates and query mining and pairing schemes to obtain higher accuracy and larger diversity of produced redescriptions.

2.4 Redescription Mining Literature Survey

The field of redescription mining was introduced in the work of Ramakrishnan et al. [3], which presents a decision tree-based redescription mining algorithm called the CARTwheels. The algorithm builds two decision trees (one for each view) that are joined in the leaves. Redescriptions are found by examining the paths (joining the conditions) from the root node of the first tree to some specified class and the paths from the root node to the matching leaf of the second tree. The algorithm uses multi-class classification to guide the search between the two views. Zaki and Ramakrishnan [11] used a lattice of closed descriptor sets to find redescriptions whereas Parida and Ramakrishnan [10] used a relaxation lattice for mining exact and approximate redescriptions. Gallo et al. [12] presented the greedy algorithm and the MID (Mining Interesting Descriptors) algorithm based on frequent itemset mining. Galbrun and Miettinen [13] extended the functionality of the greedy approach by Gallo et al. [12] to allow creating redescriptions from data containing numerical attributes. Galbrun and Kimming extended redescription mining to a relational [25] setting, while Galbrun and Miettinen [15] made extensions that enable interactive redescription mining. Two tree-based algorithms were proposed by Zinchenko [14], [38]. These approaches use decision trees in a non-Boolean setting and present two different tree-based methods for redescription mining. The first method uses layer-by-layer tree construction, while the second method uses decision trees of different depths in the redescription construction process.

Galbrun and Miettinen introduced a tool Siren [15] that enables exploration and analysis of redescriptions. The visualization and analyses techniques provided by the tool are mostly aimed towards the analyses of individual redescriptions. The tool is capable of visualizing geographical locations described by some redescription for appropriate datasets. Kalofolias et al. [39] introduced a method aimed at removing redundant redescriptions from the produced redescription sets.

Redescription mining has been applied in several different domains. Ramakrishnan et al. [3] used their redescription mining algorithm, CARTwheels in a biological problem of characterizing similarities and differences in yeast gene expression behaviour across related families of stresses. Parida and Ramakrishnan [10] applied redescription mining to data obtained from six different organisms: Baker's yeast, plant Arabidopsis, Worm Fly, Mouse and human with the goal of understanding the connection between biological processes and their location in a cell. The ultimate goal was to gain understanding that may help in function assignment for unassigned genes. The second important goal was to understand specific constructs in Eukaryotic organisms by observing redescriptions constructed using data from all six organisms. Zaki and Ramakrishnan [11] used interactive redescription mining to explore the gene expression datasets from micro-array experiments conducted on the yeast Saccharomyces cerevisiae, introduced in [3]. Additional applications of redescription mining were performed by Ramakrishnan and Zaki [40] to create redescriptions of genes contained in Saccharomyces cerevisiae using information about gene expression, gene functions (contained within GO ontology) and clusters of time course datasets. They also used redescriptions to describe genes using physiological indicators and activated pathways, modelled gene regulatory network (activations and deactivations of genes are used as descriptors) and perform cross-taxonomic and cross-genomic comparisons (the goal is to related genes described by functional anotations obtained from two different ontological contexts). Gallo et al. [12] used three different datasets *Courses, Web, and DBLP* to evaluate their redescription mining algorithms. Courses data contains students' course enrolment data of CS students at the University of Helsinki. The web data contains information about web pages of computer science departments of United States universities (terms occurring on web pages and terms occurring on hyperlinks pointing to the web pages). The DBLP dataset contains information about co-authorships and information about the conferences on which these authors published papers. All previously mentioned applications were performed on data sets containing only Boolean attributes.

The DBLP data set is also used by Galbrun [9] to evaluate the performance of the ReReMi algorithm. Evaluation of this algorithm with numerical attributes was performed on the dataset containing numerical attributes describing bioclimatic envelopes (the bioclimatic conditions required for a species to survive)—called Bio. This data set contains the information about the habitats of different mammalian species on different locations across Europe and the information about the corresponding weather conditions. Relating these information and finding bioclimatic envelopes (also called niches) can help predict the impact of global warming (see [9], [41]). Galbrun and Miettinen [42] use redescription mining to analyse political opinions. More specifically, they related the socio-economical background of voters with their political stance.

2.5 Data Mining Fields Related to Redescription Mining

Redescription mining is related to several other fields of data mining. The most obvious relation exist with association rule mining [28]–[30], two-view data association discovery [43], clustering [44]–[48] and its special form conceptual clustering [49], [50]. Subgroup discovery [51]–[54], emerging patterns [53], [55], contrast set mining [53], [56] and exceptional model mining [57] share the goal of finding descriptive rules describing entities, but additionally use information about target class in rule construction to discover mutually different types of patterns. The relationship, based on the type of task, between aforementioned fields can be seen in Figure 2.5.

Supervised tasks, such as single-label or multi-label classification, have different goals from redescription mining (to accurately predict target variables using descriptive variables). However, these tasks can be used as a redescription mining problem in which target variables are used as a separate descriptive view. Such procedure may increase the understanding of different target variables and their connection to different descriptors.

Association rule mining [28]–[30] finds pairs of queries (in a form of one-directional associations) describing sets of entities and revealing associations between different attributes used in these queries. As opposed to association rule mining that finds one-directional associations, associations discovered by redescription mining are bi-directional. The goal of two-view data association discovery [43] is to find a small, non-redundant set of associations that explain the relationship between two views. To achieve its goals, it produces both uni-directional and bi-directional associations.

Clustering discovers groups of similar instances with respect to a set of attributes. However, these groups do not necessarily have straightforward descriptions which are understandable to the user. A step towards resolving this problem is made through conceptual clustering [49], [50]. Algorithms from this data mining task find clusters and concepts that describe them. Redescription mining discovers clusters that are described by at least two different concepts. Clustering can also be used on data containing multiple views through multi-view [58], [59] and multi-layer clustering [4] to find groups of entities that are strongly connected across multiple views. The goal of obtaining groups that have similar properties

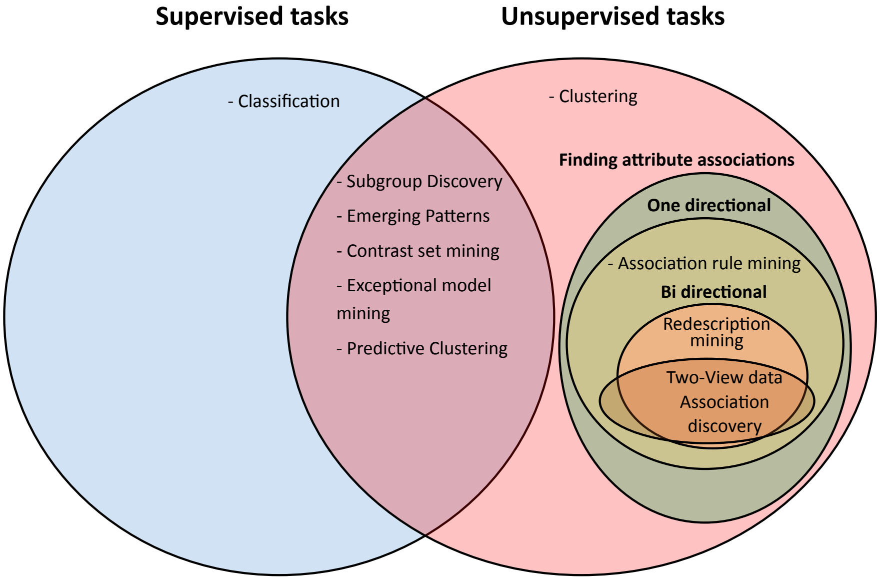


Figure 2.5: Relation between redescription mining and other related tasks.

across multiple-views is common both to multi-view clustering and redescription mining. However, redescription mining searches for such groups that can be accurately described using a predefined query language. This may result in differences in produced groups. For instance, strict query language may cause different clusters to be segmented in various strongly connected sub-clusters.

Subgroup discovery [51], [52] finds queries describing groups of instances having unusual and interesting statistical properties with respect to the target variable, which are often unavailable in purely descriptive tasks. Exceptional model mining [57] extends subgroup discovery to more complex target concepts. It searches for subgroups such that a model trained on this subgroup is exceptional based on some property.

Emerging Pattern Mining [55] searches for patterns with some discriminative properties (whose support siginficantly changes from one class or dataset to another), while Contrast Set Mining [56] identifies monotone conjunctive queries that best discriminate between instances containing one target class from all other instances.

Redescription mining does not explicitly use information about target variables, however they may be used as additional view, which would allow finding bi-directional associations between different subsets of attributes and target variables.

2.6 Multi-view approaches related to redescription mining

In this section we explain similarities and general differences between redescription mining algorithms and algorithms for solving different multi-view learning and multi-view kernel tasks.

Multi-view learning [60] is a broad field containing multiple different tasks such as: *multi-view dimensionality reduction*, *multi-view semi-supervised learning*, *multi-view supervised learning*, *multi-view active learning*, *multi-view ensemble learning*, *multi-view clustering* etc.

The main similarity shared between algorithms solving aforementioned tasks and redescription mining algorithms is that they are able to use multiple views to improve the accuracy or confidence in the obtained result. The main difference between redescription mining algorithms and other algorithms for multi-view or multi-kernel learning is that redescription mining aims to find groups that can be re-described using one or more disjoint views using a predefined query language. Other aforementioned tasks do not share this goal but have various different objectives.

Multi-view supervised and semi-supervised learning methods use multiple views to improve the predictive power of underlying models. A slight similarity exists between algorithms for multi-view semi-supervised learning [60] that follow a scheme called *co-training* and tree-based redescription mining algorithms. Semi-supervised algorithms with co-training use predictions produced by a model on one view (for the top highly confident entities) to enhance the training (by enlarging the training set) of the model on a view where target labels for these entities are missing. When training tree-based models in redescription mining algorithms, information about the model created on one view is used to guide the construction of a model on the other view.

Multi-view clustering approaches [58], [59] require grouping the same subset of entities in each view, which is also the goal of redescription mining. Depending on the type of clustering, this is achieved in different ways: common eigenvector matrix, using Laplacian eignevector produced in one view to cluster samples and use this clustering to modify Laplacian on other view, co-regularization, common coefficient matrix and common indicator matrix. However, these approaches do not offer descriptions or re-descriptions of obtained clusters.

Multi-layer clustering approach [4] uses rules to create clusters shared by multiple-views. However, this approach does not produce descriptions of produced clusters, although it can provide information about most important features associated with a particular cluster.

Multi-view dimensionality reduction methods aim to reduce the dimensionality of data taking into account information about available views. Redescription mining can potentially be used to achieve similar goals, however to our knowledge no such attempt has been made so far.

Redescription mining can be performed in a setting similar to active learning. This setting is called interactive redescription mining. The user selects the most interesting redescription with possibility of further expansions and more detailed analyses. With the extensions proposed in this thesis (see Contribution 2 [20]) it is possible to use ensembles of redescription mining algorithms to create redescriptions. As in multi-view ensemble learning, it is expected that larger diversity of algorithms contained within ensemble should produce sets of superior properties.

Multi-view kernel methods [61] extend different kernel-based methods (such as SVM, SVR) to allow learning from multiple views. These techniques have the largest connection to tree-based redescription mining algorithms in a sense that a model for each view is learned. Though techniques to combine these models and the overall goals differ significantly. Redescription mining is a descriptive task whereas multi-view kernel methods aim at predicting some target concept using multiple views.

Chapter 3

Redescription Evaluation Measures

In this chapter, we first explain some related quality measures designed in several fields related to redescription mining, then we define and explain different redescription evaluation measures grouped by their type.

3.1 Rule Evaluation Measures

Fields related to redescription mining, such as rule mining, subgroup discovery, association mining etc., use different measures to evaluate the quality of discovered patterns. Some of these measures can be adopted to be used to evaluate redescriptions. Because of this, we shortly summarize the main rule evaluation measures used in the data mining fields related to redescription mining, described in Section 2.5, and motivate our choice for introducing redescription redundancy and complexity measures into the redescription mining process.

3.1.1 Rule learning

Depending on the type of supervised task (classification or regression), rule sets are evaluated by computing *classification error*, *Relative root mean squared error*, *Pearson's Correlation coefficient* and *Complexity* [37]. A more detailed overview can be found in [62]. Many other classification, regression and correlation measures can be used to assess the quality of a produced rule set in the presence of a target variable [63]. The most commonly used measures include accuracy, precision, recall, the true positive rate (TPR), the false positive rate (FPR), the F-measure, the area under the ROC curve (AUC) and the area under the precision-recall curve (AUPRC).

Since redescription mining is an unsupervised task, measures that inherently use information about the target label, such as classification error, relative root mean squared error, AUC etc. are not used to evaluate redescriptions or redescription sets.

In the field of rule learning, some of the important individual rule evaluation measures that do not use information about target labels are *dispersion*, *coverage*, *distance to existing rules* and *prototype dissimilarity* [37].

The *prototype vector* of a set of entities A for an attribute a_j is defined as $p_A(\vec{a_j}) = (\frac{n_1}{|A|}, \frac{n_2}{|A|}, \ \ldots, \ \frac{n_L}{|A|})$, where L denotes the number of possible different values of attribute a_j, and n_k the number of entities with attribute value l_k. If $A = \{e_s\}$ and $a_j(e_s) = l_p$, then $n_j = 0, \ j \neq p$, $n_p = 1$ and $|A| = 1$. In case of numerical attributes, mean of vectors contained in a set can be used as a prototype vector.

The *dispersion* (also known as variability) measures the variation of values of entities contained in the support set of a given rule for attributes constructing the rule. The attributes can be weighted, which allows for expressing the importance of a particular subset of attributes. If we use an arbitrary vector distance measure d, the dispersion is defined as: $disp(A, a_j) = \frac{1}{2|A|}\frac{L}{L-1}\sum_{i=1}^{|A|} d(p_{e_i}, p_A)$. In case of numerical variables, variance can be used as a dispersion measure. Similar distribution value analysis of individual redescriptions is available in our redescription exploration framework [21].

The *coverage* measures the fraction of entities described by a rule. Thus, $cov(r, E) = \frac{|supp(r)|}{|E|}$, where r denotes a rule and E a dataset containing $|E|$ entities. Entities can be weighted to express the importance of a given subset of entities. In this setting, rule coverage is defined as $cov(r, E, \vec{w}) = \frac{\sum_{e_i \in supp(r)} w_i}{\sum_{e_i \in E} w_i}$. Similar measure, modified to assess unique redescription coverage is used in our redescription mining algorithm [18].

The *distance to existing rules* is defined as the average distance to each rule contained in some rule set $\mathcal{R}'$. The pairwise rule distance is defined as $d(r_j, r_k) = \frac{1}{|E|}\sum_{i=1}^{|E|} d_1(r_j, r_k, e_i)$, where:

$$d_1(r_j, r_k, e_i) = \begin{cases} 0, & e_i \in supp(r_j) \cap supp(r_k) \\ 0, & e_i \notin supp(r_j) \cup supp(r_k) \\ 1, & \text{otherwise} \end{cases}$$

The alternative way of assessing the rule dissimilarity is by computing the Jaccard similarity coefficient between the support sets of two different rules $(J(supp(r_j), supp(r_k)))$. We use Jaccard index as a similarity measure in redescription mining (see Section 3.4).

The *prototype dissimilarity* measures the difference between a prototype vector of a set of entities contained in a support set of a given rule and a prototype vector obtained on all entities from the dataset [37] based on values of all attributes contained within a rule. It is used to produce rules with different prototypes from a default prototype (obtained on all entities) which increases the amount of information contained within a rule set.

The model *complexity* is defined as the number of rules in the produced rule set, $comp1(\mathcal{R}) = |\mathcal{R}|$ but can also be defined as the sum of individual complexity of each rule (number of attributes—tests contained in the rules from the rule set $\mathcal{R}$) $comp2(\mathcal{R}) = \sum_{r_i \in \mathcal{R}} |attr(r_i)|$.

Measures of redescription and redescription set complexity have an important role in redescription set construction (see Section 3.5).

A set of evaluation measures has been presented in [64] designed to evaluate rules of a form $r = B \rightarrow H$, where H denotes a set of instances for which the head of a rule is true and B denotes the set of instances for which the body of a rule is true. Complements of this set are denoted $\overline{H} = E \setminus H$ and $\overline{B} = E \setminus B$.

This general rule form occurs in predictive rules (classification/regression), subgroup discovery, some cases of exceptional model mining, contrast set mining (where one or more target variables form the head of a rule) and the unsupervised task of association rule mining (where the rule consequent has a role of the head of a rule). These evaluation measures are based on sample relative frequencies which are interpreted as probabilities. Similar measures might also be applicable for redescription mining.

Measures defined in [64] rely on the computation of the contingency table (see Table 3.1). For instance, the *accuracy* of a rule is defined as $acc(r) = p(H|B)$, the *negative reliability* as $negrel(r) = p(\overline{H}|\overline{B})$, the *sensitivity* as $sens(r) = p(B|H)$, the *specificity* as $spec(r) = p(\overline{B}|\overline{H})$, the *coverage* as $cov(r) = p(B)$, the *support* as $s_r(r) = \frac{|supp(r)|}{|E|} = p(H \cap B)$, the *novelty* as $nov(r) = p(H \cap B) - p(H) \cdot p(B)$ etc. For the complete list of such rule evaluation measures see [64].

Table 3.1: A contingency table for the rule $r = B \rightarrow H$.

	B	$\overline{B}$	
H	$\lvert H \cap B\rvert$	$\lvert H \cap \overline{B}\rvert$	$\lvert H\rvert$
$\overline{H}$	$\lvert \overline{H} \cap B\rvert$	$\lvert \overline{H} \cap \overline{B}\rvert$	$\lvert \overline{H}\rvert$
	$\lvert B\rvert$	$\lvert \overline{B}\rvert$	$\lvert E\rvert$

3.1.2 Subgroup discovery

Rules in subgroup discovery have a form $r = B \rightarrow Class$, thus they are a special case of rules $B \rightarrow H$ defined in the previous section. In subgroup discovery, the body of a rule is called a *condition*. All measures, defined in the previous section, for $B \rightarrow H$ type of rules can be applied and used to evaluate rules obtained by subgroup discovery algorithms.

Measures for rule complexity, coverage and support are also used in subgroup discovery. The *coverage* of a rule set is defined as $cov(\mathcal{R}) = \frac{1}{|\mathcal{R}|}\sum_{r_i \in \mathcal{R}} cov(r_i)$ [65]. Similarly, the *support* of a rule set is defined as $s_{SD}(\mathcal{R}) = \frac{|supp(\mathcal{R})|}{|E|} = \frac{1}{|E|}\sum_{Class_j} |Class_j \cap (\cup_{B_i \rightarrow Class_j} B_i)|$. The rule set support measures the fraction of correctly classified entities from the dataset by at least one rule. Similarly, the redescription set coverage measures the fraction of entities described by at least one redescription (see Section 3.6). The size of a rule set is defined as the $comp1(\mathcal{R}) = |\mathcal{R}|$. Many other subgroup discovery rule evaluation measures can be found in [65].

Abudawood and Flach [66] present three additional subgroup discovery rule evaluation measures: the *mutual information*, the *Chi-Squared* and the *Gini split*. The Chi-Squared test can be applied to compute the statistical dependence between the rule and the discovered target variable. Computation of theoretical statistical significance of redescriptions is based on similar principles (dependence or independence of redescription queries) but is computed from Binomial or Hypergeometric distribution (see [9] and Section 3.3). Description of several subgroup discovery rule evaluation measures, such as area under the ROC curve, the weighted relative accuracy, can be found in [65].

3.1.3 Exceptional model mining

Exceptional model mining extends the goals of subgroup discovery to different types of models and target concepts. The methodology can be applied to correlation, regression and classification models [57]. Thus, the rules can have a general descriptive form $r = B$ or a form $r = B \rightarrow H$, depending on the type of model addressed in the task.

For correlation models, three different rule evaluation measures are used. The *difference of correlation* $crd(r) = |corr(B) - corr(\overline{B})|$, where $corr(B)$ denotes (some) correlation coefficient computed for a pair of attributes on entities contained in the set B and the complement $\overline{B}$. The *entropy weighted difference of correlation* $wcrd(r) = H(p) \cdot |corr(B) - corr(\overline{B})|$ uses entropy $H(p)$ of a split between sets B and $\overline{B}$ to weigh the difference between correlation scores of selected attributes in B and $\overline{B}$. The *significance of correlation difference* is computed by applying the Fisher's z-transform and computing the p-value from the Normal distribution (for more information see [57]). The final score is computed as $1 - p$. These scores can also be used to evaluate redescriptions.

In the case of regression models, one can use the *significance of slope difference* to asses the difference between a regression model fitted to a group B and a regression model fitted to a group $\overline{B}$. The significance of the test statistics can be approximated by the t distribution (see [57], [67]). The final score is computed as $1 - p$.

Several measures can be used to assess the performance of classification models on a selected subgroup of entities. The *BDeu score* measures the predictability of a given subgroup while the Hellinger distance measures the distance between two probability distributions. It can be used to assess the distance between the probability estimate of predicting the given category of a target variable for a set of entities contained within a subgroup and the estimated probability of predicting the same category for entities not contained in the chosen subgroup.

3.1.4 Contrast set mining

A contrast set [53], [56], [68], [69] is a rule of a form $r = c_{a_1}^{(1)} \wedge \cdots \wedge c_{a_k}^{(k)}$, where $c^{(i)}$ denotes conditions on attributes a_i contained in the set of attributes $\mathcal{A}$. The rule can be defined on a set of groups $G_1, G_2, \ldots, G_n$, where $G_i \cap G_j = \emptyset,\ \forall i, j$. The main goal of Contrast Set Mining is to find all contrast sets with very different supports on different groups G_i. The *support* of a contrast set r for a group G is defined as the fraction of entities from G that are covered by contrast set r, $s_{CS}(r, G) = \frac{|supp(r) \cap G|}{|G|}$.

Two main measures are used in Contrast Set Mining: the *difference of support* and the *difference of probability.* The difference of support is defined as $\mathrm{suppdiff}(r, G_i, G_j) = max_{i,j}|supp_{CS}(r, G_i) - supp_{CS}(r, G_j)| > \delta$, where δ represents a user-defined parameter called *the minimum support difference.* For two disjoint groups $G_i,\ G_j$, the Contrast Set Mining algorithms search for pairs of subgroups that have different probabilities $p(r|G_i)$ and $p(r|G_j)$.

3.1.5 Association rule mining

Rules in association rule mining [28] are of a form $r = X \rightarrow Y$, where $X \subseteq \mathcal{A}$ and $Y \subseteq \mathcal{A}$. In contrast to the rules presented in the previous sections, association rules discover implication relation between two sets of descriptive attributes. Many different association rule measures exist in the literature. A detailed survey is presented in [70].

Widely used objective measures in association rule mining include the *support*, the *(relative) confidence*, the *lift*, the *conviction*, the *leverage*, the *improvement*, the *multiplicative improvement*, the *validity*, the *bi-lift*, the *bi-improvement* and the *bi-confidence* [71], [72]. The majority of these measures can also be applied in redescription mining. For a redescription $R = (q_1, q_2)$, the association rule measures can be applied to two implication rules $q_1 \rightarrow q_2$ and $q_2 \rightarrow q_1$, while some measures such as confidence and lift can be applied directly to evaluate redescriptions.

Below, we provide definitions of some important association rule mining measures:

The *support*: $s_{AR}(r) = \frac{|supp(r)|}{|E|} = \frac{|supp(X) \cap supp(Y)|}{|E|}$.

The *confidence*: $conf(r) = \frac{|supp(X) \cap supp(Y)|}{|supp(X)|}$.

The *relative confidence*: $rconf(r) = \frac{|supp(X) \cap supp(Y)|}{|supp(X)|} - \frac{|supp(Y)|}{|E|}$.

The *lift*: $lift(r) = \frac{|E| \cdot |supp(X) \cap supp(Y)|}{|supp(X)| \cdot |supp(Y)|}$.

The *conviction*: $conv(r) = \frac{|supp(X)| \cdot |supp(\overline{Y})|}{|E| \cdot |supp(X) \cap supp(\overline{Y})|}$.

The *leverage*: $lev(r) = \frac{|supp(X) \cap supp(Y)|}{|E|} - \frac{|supp(X)| \cdot |supp(Y)|}{|E|^2}$.

Definitions of the extended set of measures can be found in [71], [72].

The subjective measures for association rule evaluation include subjective factors that are dependent on the application field. These measures are not in the focus of this thesis, thus we refer the interested readers to a survey [71].

3.2 Redescription Accuracy

Redescription accuracy is measured as a similarity of support sets of its constituting queries. As stated in [9], many different measures, such as *matching number*, *matching ratio*, *Russel & Rao coefficient*, *Jaccard index*, *Dice coefficient*, *Rogers & Tanimoto coefficient* can be used to compute similarities between queries (see pp. 34,35 in [9]).

3.2.1 Measuring redescription accuracy with Jaccard index

The similarity of redescription queries is typically computed using the *Jaccard index.* The main reason for this is that the definition of Jaccard index treats sets symmetrically, which is a preferable property for redescription evaluation measures. The Jaccard index has been used before in different rule learning and mining fields, most notably in association rule mining [70].

Definition 3.1. For a redescription $R = (q_1, q_2)$, the *Jaccard index* (Jaccard similarity coefficient) is defined as:

$$J(R) = \frac{|supp(q_1) \cap supp(q_2))|}{|supp(q_1) \cup supp(q_2)|}$$

The Jaccard index equally penalizes entities described by either query that are not described by the other, and emphasises the importance of entities described by both queries. Some measures also take into account the entities that are not described by either query, however this set is of little interest in redescription mining, since we evaluate a redescription of a given set of entities.

3.2.2 Measuring redescription accuracy with Jaccard index in the presence of missing data

Evaluating redescription accuracy is much harder if the underlying data contains missing values. The main reason is that, in some cases, it is not possible to determine if some entities are described by a query or not. For a redescription $R = (q_1, q_2)$, we use the notation from [9] to denote $E_{1,1}$—a set of entities described by both queries, $E_{1,0}$—a set of entities described by the first query but not described by the second query, $E_{0,1}$—a set of entities described by the second query but not described by the first query, $E_{0,0}$—a set of entities that are not described by either query, $E_{?,1}$—a set of entities for which it is not possible to determine if they are described by the first query due to missing values and described by the second query, $E_{1,?}$—a set of entities described by the first query but for which it is not possible to determine if they are described by the second query. A set $E_{?,?}$ contains entities for which it is not possible to determine if they are described by either query due to missing values.

Depending on the way in which we treat the missing values, several versions of the Jaccard index can be defined:

- *Rejective Jaccard index*: $J_{rej}(q_1, q_2) = \frac{|E_{1,1}|}{|E_{1,1}|+|E_{1,0}|+|E_{0,1}|}$, thus we completely ignore the missing values. This measure gives relatively accurate estimates if there is a very small amount of missing values in the data.

- *Optimistic Jaccard index*: $J_{opt}(q_1, q_2) = \frac{|E_{1,1}|+|E_{?,1}|+|E_{1,?}|+|E_{?,?}|}{|E_{1,1}|+|E_{?,1}|+|E_{1,?}|+|E_{?,?}|+|E_{0,1}|+|E_{1,0}|}$. The optimistic Jaccard index considers all entities, for which the membership to a query support set can not be determined due to missing values, to be contained in the redescription support set.

- *Pessimistic Jaccard index*: The pessimistic Jaccard index considers all entities, for which the membership to a query support set can not be determined due to missing values, not to be contained in the redescription support set. $J_{pess}(q_1, q_2) = \frac{|E_{1,1}|}{|E_{1,1}|+|E_{?,1}|+|E_{1,?}|+|E_{?,0}|+|E_{0,?}|+|E_{?,?}|+|E_{0,1}|+|E_{1,0}|}$.

- *Query non-missing Jaccard index*: $J_{qnm}(q_1, q_2) = \frac{|E_{1,1}|}{|E_{1,1}|+|E_{?,1}|+|E_{1,?}|+|E_{0,1}|+|E_{1,0}|}$. The main motivation for this measure came from earlier work on rule learning and association rule mining, where the support set of a rule does not contain entities that can-not be evaluated due to missing values (a study considering missing values in association rule mining can be seen in [73]). This led to the development of a novel measure which, unlike the rejective Jaccard index, takes into account the distribution of missing values. The newly defined measure evaluates as positive only those entities that are described by both redescription queries, as opposed to the optimistic Jaccard index that expands the redescription support set with entities for which this is not certain. It obviously holds that $J_{pess}(q_1, q_2) \leq J_{qnm}(q,q_2) \leq J_{rej}(q_1, q_2)$. By analytically computing $J_{opt} - J_{qnm}$, we see that:

$$J_{opt} - J_{qnm} = \frac{(|E_{?,1}| + |E_{1,?}|) \cdot (|E_{1,1}| + |E_{0,1}| + |E_{1,?}| + |E_{?,1}| + |E_{1,0}|)}{y}$$

  where $y = (|E_{1,1}| + |E_{?,1}| + |E_{1,?}| + |E_{?,?}| + |E_{0,1}| + |E_{1,0}|) \cdot (|E_{1,1}| + |E_{?,1}| + |E_{1,?}| + |E_{0,1}| + |E_{1,0}|)$. Thus, $J_{qnm} \leq J_{opt}$.

The first three Jaccard index variants were presented in [13]. The fourth measure is an original contribution presented in [19].

From the analysis presented for the query non-missing Jaccard index, several very important properties of the measure can be computed. The first property is $\lim_{|E_{1,1}|\rightarrow\infty}(J_{opt} - J_{qnm}) = 0$. From $(J_{qnm} - J_{pess}) = \frac{(|E_{1,1}|)\cdot(|E_{0,?}|+|E_{?,0}|+|E_{?,?}|)}{y_1}$, where $y_1 = (|E_{1,1}| + |E_{?,1}| + |E_{1,?}| + |E_{?,?}| + |E_{0,1}| + |E_{1,0}| + |E_{?,0}| + |E_{0,?}|) \cdot (|E_{1,1}| + |E_{?,1}| + |E_{1,?}| + |E_{0,1}| + |E_{1,0}|)$, it follows that $\lim_{|E_{1,1}|\rightarrow\infty}(J_{qnm} - J_{pess}) = 0$, thus the measure has very close values to the pessimistic and optimistic Jaccard index when the number of correctly described entities is much larger than the number of entities that can- not be evaluated due to missing values or being described by only one query.

It is also interesting to note that $\lim_{|E_{?,?}|\rightarrow\infty}(J_{opt} - J_{qnm}) = \frac{1}{|E_{1,1}|}$, thus the measure will be much more conservative than the optimistic Jaccard index, though the difference diminishes with the size of redescription support set. Similarly $\lim_{|E_{?,?}|\rightarrow\infty}(J_{qnm} - J_{pess}) = \frac{|E_{1,1}|}{|E_{1,1}|+|E_{?,1}|+|E_{1,?}|+|E_{0,1}|+|E_{1,0}|}$. In the presence of a high number of missing data, the difference between the query non-missing and pessimistic Jaccard index will be large if redescriptions contain larger support sets. However, it will be very small if the size of a set of entities described by at least one query or containing missing values for at least one query dominates the size of a redescription support set.

If a high number of entities exists described by one query which can-not be evaluated by the other due to missing values, the following holds: $\lim_{|E_{?,1}|\rightarrow\infty}(J_{opt} - J_{qnm}) = 1$, thus the query non-missing Jaccard index strongly differs from optimistic Jaccard on such redescriptions. Similarly, $\lim_{|E_{?,1}|\rightarrow\infty}(J_{qnm} - J_{pess}) = 0$. In the case where the number of entities contained in the sets $E_{0,?}$ and $E_{?,0}$ is very large, the increase in size of such sets does not affect the difference between pessimistic and query non-missing Jaccard index, since both measures consider that those entities are not a part of the redescription support set (similar as the $E_{0,0}$ set). However, $\lim_{|E_{?,0}|\to\infty}(J_{qnm} - J_{pess}) = \frac{|E_{1,1}|}{|E_{1,1}|+|E_{?,1}|+|E_{1,?}|+|E_{0,1}|+|E_{1,0}|}$, as when $|E_{?,?}| \to \infty$.

These results reflect the intuition that highly accurate redescriptions whose support set size dominates the number of entities, whose membership in the redescription support set can not be determined, due to missing values in one or both queries most probably describe some specific properties of the entities. In such cases, it is more probable that the entities from the set $E_{0,?}$ and $E_{?,0}$ truly belong to the set $E_{0,0}$ and the entities contained in the set $E_{?,?}$ would be correctly assigned to $E_{0,0}$ or $E_{1,1}$ depending on the real values for the needed attributes. Thus in such cases, the query non-missing measure is designed to behave more like the optimistic than the pessimistic Jaccard index. On the other hand, if the observed redescription contains a large number of entities that are described by only one query but not the other (possibly due to missing values), then the measure behaves as the pessimistic Jaccard index.

The redescription *variability index*, defined in [20], is defined as $variability(R) = J_{opt}(R) - J_{pess}(R)$. Using this measure allows finding redescriptions whose accuracy is unaffected by the missing values present in the data. This measure can be used when it is important to analyse highly accurate redescriptions that are unaffected by missing values, thus it is expected for them to remain accurate even when new information is incorporated into the data (i.e some missing values are replaced by newly obtained data).

3.3 Statistical Significance of Redescriptions

Since it is relatively easy to obtain redescriptions with large support, for which it is highly probable to have a very high overlap of their queries, additional redescription evaluation measures are required. Thus, it is preferred to have redescriptions that reveal some more specific knowledge about the studied problem that is harder to obtain by random sampling from the underlying data distribution. This property is measured by the statistical significance (p-value) for each obtained redescription.

As described in [9], there are two ways to theoretically measure the statistical significance of a redescription:

- Estimate the probability of obtaining a pattern of a given size or larger if two independent queries with the same marginal probabilities, as observed, are combined into a redescription. The p-value in this case is computed from the binomial distribution as:

$$p(q_1, q_2) = \sum_{n=|o|}^{|E|} \binom{|E|}{n} (p_1 \cdot p_2)^n \cdot (1 - p_1 \cdot p_2)^{|E|-n}$$

  where the marginal probability of a query q_1 is denoted as $p_1 = \frac{|supp(q_1)|}{|E|}$ and a marginal probability of a query q_2 is denoted as $p_2 = \frac{|supp(q_2)|}{|E|}$. The set of entities in the intersection of the queries is denoted as $o = supp(q_1) \cap supp(q_2)$.

- Compute the probability that two randomly chosen sets of cardinalities $|supp(q_1)|$ and $|supp(q_2)|$ have an overlap of cardinality $|o|$ or larger. The p-value is computed from the hypergeometric distribution as:

$$p(q_1, q_2) = \sum_{n=|o|}^{|E|} \frac{\binom{|supp(q_1)|}{n}\binom{|E|-|supp(q_1)|}{|supp(q_2)|-n}}{\binom{|E|}{|supp(q_2)|}}$$

Both estimates are optimistic when the assumption that all entities can be sampled with equal probability does not hold (which is often the case in practice).

We used the first definition of statistical significance throughout this thesis.

3.4 Redescription Redundancy

We have defined (presented in [19]) two redescription quality measures based on properties of a redescription set that contains them. These measures provide information about the level of redundancy of a given redescription, with respect to the described entities and attributes used in redescription queries, compared to other redescriptions contained in the given redescription set. The measure providing information about the redundancy of entities contained in the redescription support is called the average redescription entity Jaccard index and is defined as:

$$AEJ(R_i) = \frac{1}{|\mathcal{R}|-1} \cdot \sum_{j=1}^{|\mathcal{R}|} J(supp(R_i), supp(R_j)),\ i \neq j$$

Analogously, the measure providing information about the redundancy of attributes contained in redescription queries, called the average redescription attribute Jaccard index, is defined as:

$$AAJ(R_i) = \frac{1}{|\mathcal{R}|-1} \cdot \sum_{j=1}^{|\mathcal{R}|} J(attrs(R_i), attrs(R_j)),\ i \neq j$$

The measures for average entity/attribute redundancy of a redescription set $\mathcal{R}$ are defined as:

$$AEJ(\mathcal{R}) = \frac{1}{|\mathcal{R}|} \cdot \sum_{j=1}^{|\mathcal{R}|} AEJ(R_j)$$

$$AAJ(\mathcal{R}) = \frac{1}{|\mathcal{R}|} \cdot \sum_{j=1}^{|\mathcal{R}|} AAJ(R_j)$$

These measures can be used to select redescriptions describing subsets of entities that are (not) described often by other redescriptions from the redescription set as demonstrated in [21]. A slightly modified version of these measures is used in [20] to allow selecting non-redundant redescriptions in the process of redescription set construction. For a redescription $R_i \in \mathcal{R} \backslash \mathcal{R}_{red}$ we compute: $score_{elemSim}(R_i) = max_j\ J(supp(R_i), supp(R_j))$, $j = 1, \ldots, |\mathcal{R}_{red}|$ and $score_{attrSim}(R) = max_j\ J(attrs(R_i), attrs(R_j)),\ j = 1, \ldots, |\mathcal{R}_{red}|$. Related measures, using entity/attribute frequencies instead of support set/attribute set overlap, are used in [18], [19] to select redescriptions with diverse redescription supports and attributes used in redescription queries. These measures, motivated by the work from subgroup discovery [65], [74], are defined as: $redScoreEl(R) = \frac{\sum_{e\in supp(R)}(elFreq[e]-1)}{\sum_{e\in E}(elFreq[e])}$, $redScoreAt(R) = \frac{\sum_{a\in attr(R)}(attrFreq[a]-1)}{\sum_{a\in \mathcal{A}}(attrFreq[a])}$, for $R \in \mathcal{R}$. They are defined similarly for $R \notin \mathcal{R}$: $redScoreEl(R) = \frac{\sum_{e\in supp(R)}(elFreq[e])}{\sum_{e\in E}(elFreq[e])}$, $redScoreAt(R) = \frac{\sum_{a\in attr(R)}(attrFreq[a])}{\sum_{a\in \mathcal{A}}(attrFreq[a])}$.

The main difference between the two presented approaches for measuring redundancy is that the first approach measures the difference of described subsets of entities/attributes used in redescription queries. Exploration of different parts of entity/attribute space is enforced by observing the distance from the redescription describing the most similar set of entities and containing the most similar set of attributes in its queries. On the other hand, the second measure aims to distribute redescriptions equally across the entity/attribute space (find redescriptions that describe different parts of entity space using diverse attributes). This is achieved by computing frequencies of entity occurrence in redescription support sets, of redescriptions currently contained within a redescription set and frequencies of attribute occurrence in redescription queries of these redescriptions. These measures consider redescriptions that describe entities/attributes with low occurrence frequency to be preferred (less redundant).

As a result of their properties, the aforementioned measures grade redescription redundancy differently. The first measure (measuring the similarity of redescription support and attribute sets) considers redescriptions describing similar subsets of entities or containing similar attributes in their queries (even though these entities/attributes may have a small occurrence frequency) to be more redundant than redescriptions describing very different subsets of entities or containing different subsets of attributes (which may have a relatively high occurrence frequency). On the other hand, the second measure considers redescriptions describing entities or containing attributes with higher frequencies to be more redundant than redescriptions describing entities with small occurrence frequency or containing attributes with small occurrence frequency (even though there may exist some redescription in the redescription set describing a very similar subset of entities or containing a very similar subset of attributes in its queries).

3.5 Redescription Complexity

We introduced a measure of redescription complexity in [19]. This measure computes the normalized size of redescription queries based on the number of attributes occurring in its queries. The normalization factor is a parameter which allows the user to tune the measure based on their application. If the user considers redescriptions containing a total of $n \in \mathbb{N}$ attributes to be highly complex, a general measure of redescription complexity is defined as:

$$R_{comp} = \begin{cases} \frac{|attr(R)|}{n} & , |attr(R)| < n \\ 1 & , n \leq |attr(R)| \end{cases}$$

This basic measure of complexity is defined with the intuition that redescriptions containing shorter queries are generally easier to understand. A more general definition of R_{comp} is:

$$R_{comp} = \begin{cases} f(|attr(R)|, n) & , |attr(R)| < n \\ 1 & , n \leq |attr(R)| \end{cases}$$

where $f : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{R}$ represents a general function evaluating complexity of a given redescription based on the number of attributes used in redescription queries and number n which denotes the minimal number of attributes constructing redescription queries, for which redescriptions are considered highly complex by the user. In such a setting, the linear function can be replaced with the logarithm ($f_{ln} = \frac{ln(\frac{|attr(R)|}{n}+1)}{ln(2)}$), the exponential function ($f_{exp} = e^{\frac{|attr(R)|}{n}-1}$), etc. However, other aspects of redescription queries, such as types of logical operators used, also affect the query and redescription complexity and can be included in this measure.

3.6 Coverage

For a set of redescriptions $\mathcal{R}$, the entity coverage $cov_e(\mathcal{R})$ is defined as a fraction of entities contained in the support set of at least one redescription from $\mathcal{R}$.

$$cov_e(\mathcal{R}) = \frac{|\{e_i \in E,\ \exists R_j \in \mathcal{R},\ e_i \in supp(R_j)|}{|E|}$$

Analogously, the attribute entity coverage $cov_a(\mathcal{R})$ is defined as a fraction of attributes contained in the queries of at least one redescription from $\mathcal{R}$.

$$cov_a(\mathcal{R}) = \frac{|\{a_i \in \mathcal{A},\ \exists R_j \in \mathcal{R},\ a_i \in attrs(R_j)|}{|\mathcal{A}|}$$

The unique coverage of a redescription R with respect to a set of redescriptions $\mathcal{R}$ is defined as:

$$cov_u(R, \mathcal{R}) = \frac{|\{e_i \in E,\ e_i \in supp(R)\ \wedge\ \nexists R_j \in \mathcal{R},\ e_i \in supp(R_j)\}|}{|E|}$$

The unique coverage is used in the redescription optimization process (see [18]) to increase the score of redescriptions describing entities that are not described by any other redescription from the redescription set $\mathcal{R}$.

Chapter 4

Redescription Mining with Predictive Clustering Trees

In this chapter we motivate the use of Predictive Clustering Trees for redescription mining, present the CLUS-RM algorithm and explain the main differences and novelties of the approach compared to the existing redescription mining algorithms based on decision-trees.

4.1 Predictive Clustering Trees

Predictive Clustering Trees (PCTs) [16], [35], [36] combine the ideas from clustering [44]–[48] and classification/regression decision trees [75] to increase the predictive performance of the learned model. In essence, PCTs can use the information about the descriptive attributes, about the target attributes or both to produce different clusters that are used to make predictions (see Figure 4.1).

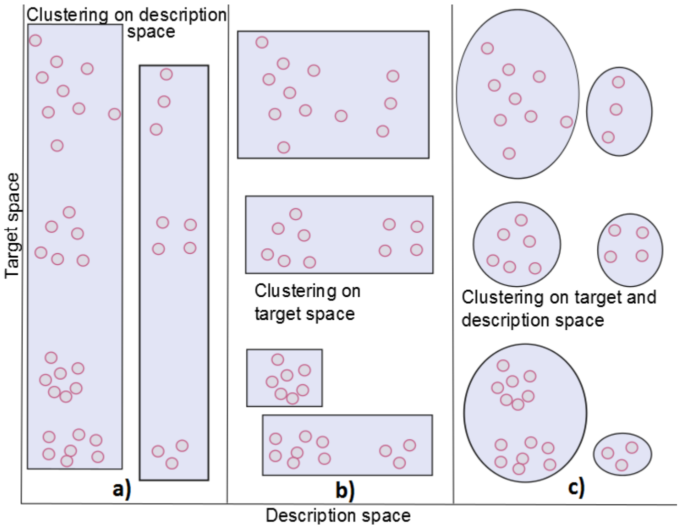

Clustering on description space
Target space
Clustering on target space
Clustering on target and description space
a)
b)
c)
Description space

Figure 4.1: Various types of object groupings: a) by using only descriptive attributes (as in unsupervised clustering), b) by using only target attributes (as in classification), c) by using target and descriptive attributes (capability of the predictive clustering). This Figure is based on Figure 2.1 from [37] and Figure 2.11 from [36].

Formally, the *cluster assignment function* has been defined as a function $f : E \times \mathbf{C} \rightarrow 2^E$ such that $\forall e_i \in E,\ \forall \mathcal{C} \in \mathbf{C}, f(e_i, \mathcal{C}) \in \mathcal{C}$. For a given clustering space $\mathbf{C}$ of a set of entities

E, the function f assigns each possible entity to a cluster in clustering $\mathcal{C}$.

Definition 4.1 (Predictive Clustering). Given a set of entities E, a distance function d defined on pairs of entities from E, a prototype function p and a cluster assignment function f, the task of predictive clustering is to find a clustering $\mathcal{C}$ over E that maximizes $Q(\mathcal{C}) = -\mathbb{E}[d(e_i, p(f(e_i, \mathcal{C})))^2]$, for $e_i \in E$. $\mathbb{E}$ denotes the expected value.

The task defined in definition 4.1 can be applied to classification and regression tasks. Given an entity space E and the prediction space P, a *target function* $\pi : I \mapsto P$ is a function mapping entities into their target values. The *predictor function* $pred_{\mathcal{C}} : E \mapsto P$ is defined as $pred_{\mathcal{C}}(e_i) = \pi(p(f(e_i, \mathcal{C})))$. *Classification* is defined as a special case of predictive clustering where the range of π is nominal and $d(e_i, e_j) = d_1(\pi(e_i), \pi(e_j))$.

$$d_1(\pi(e_i), \pi(e_j)) = \begin{cases} 1 & , \pi(e_i) = \pi(e_j) \\ 0 & , \pi(e_i) \neq \pi(e_j) \end{cases}$$

Similarly, *regression* is a special form of predictive clustering where the range of π is continuous and $d(e_i, e_j) = d_2(\pi(e_i), \pi(e_j))$. Many regression models minimize the *mean squared prediction error* [76].

The PCTs are induced in the standard top-down form [16], [17], [35], [36], already used for the construction of decision trees [75]. The PCT induction process is presented in Table 4.1. The Algorithm *PCT* defines the top-down induction of the PCT. The procedure starts by finding the best test to split the entities from set E. The test is an attribute-value pair that determines how the data is split into smaller subsets. The procedure first finds the best split on the entire dataset and then recursively splits these subsets into smaller subsets until some termination criteria is reached (for example maximum tree depth, maximal number of entities in the node etc.). The prototype function that is calculated at each leaf returns the tuple, with the mean values in case of numerical or majority class in case of nominal target variables, as prediction. The mean (majority) values are calculated using the training instances that belong to the given leaf $E_k \subseteq E$.

Table 4.1: The top-down induction algorithm for PCT [17].

```
Algorithm 4.1: PCT
  Input: A dataset $E$
  Output: A Predictive
          Clustering tree

  $(t^*, h^*, P^*) = BestTest(E)$;
  if $t^* \neq none$ then
      foreach $E_i \in P^*$ do
          $tr_i = PCT(E_i)$;
      end
      return node($t^*, \cup_i tr_i$);
  else
      return
       leaf(Prototype($E_k$));
  end
```

```
Algorithm 4.2: BestTest
  Input: A dataset $E$
  Output: The best test ($t^*$), its heuristic score
          ($h^*$) and the partition ($P^*$) it induces
          on the dataset ($E$)
  $(t^*, h^*, P^*) = (none, 0, \emptyset)$;
  foreach possible test $t$ do
      $P$ = partition induced by $t$ on $E$;
      $h = Var(E) - \sum_{E_i \in P} \frac{|E_i|}{|E|} Var(E_i)$;
      if $h > H^* \wedge Acceptable(t, P)$ then
          $(t^*, h^*, P^*) = (t, h, P)$;
      end
  end
  return $(t^*, h^*, P^*)$;
```

The Algorithm *BestTest* determines the best test (attribute-value) pair for a given subset of the input dataset, or the whole dataset for the first split. The procedure returns the best test, the heuristic score of this test and the partition obtained after splitting the input subset E_k into smaller subsets. The partition is determined by computing the variance reduction heuristics as defined in the Algorithm *BestTest*. $Var(E) = \frac{1}{|E|}\sum_{e_i \in E} d(L_i, \overline{L_i})^2$, where L_i denotes the class vector of entity e_i and $\overline{L}$ denotes the mean class vector of a set of entities E. The procedure can be extended to compute splits for different types of target attributes such as: single (multiple) target classification and regression, and hierarchical multi label classification targets [16], [17], [35], [36]. These types of targets are called structured outputs [17]. In the same work [17], Kocev et al. explore the use of ensemble models for structured output prediction.

For multi-target regression problems, where $\mathcal{Y}$ denotes the set of target variables and $|\mathcal{Y}| = T$, the splitting heuristics is computed as $Var(E) = \sum_{i=1}^{T} Var_{Y_i}(E)$. Variances for each set of target variables are normalized to have equal impact on the final variance score. Multi-target classification heuristics is computed as $Var(E) = \sum_{i=1}^{T} Gini_{Y_i}(E)$. The sum of entropies can also be used as a heuristics $Var(E) = \sum_{i=1}^{T} Entropy_{Y_i}(E)$. In case of hierarchical multi-label classification, the distance function in the variance computation formula is weighted by the depth of the class in the class hierarchy, thus $d_h(L_1, L_2) = \sqrt{\sum_{l=1}^{|L|} w(c_l) \cdot (L_{1,l} - L_{2,l})^2}$.

In our work [18]–[20], we use the PCTs ability to perform the multi-target regression and multi-label classification to produce rules, which are used to produce highly accurate, diverse redescriptions. Using multi-target PCTs allows more effective guided search by creating one model to find multiple redescriptions using nodes at all levels of the tree. Further, it has been shown [77] that due to the property of inductive transfer, the multi-target trees can outperform single-class classification or regression trees.

4.2 Related Tree-Based Redescription Mining Approaches

By their construction, redescription mining approaches can be divided into: a) itemset-based, b) greedy and c) tree-based redescription mining algorithms.

In this section, we describe the existing tree-based redescription mining approaches which use decision trees [75] to construct redescriptions. These approaches are methodologically related to the CLUS-RM algorithm (presented in Section 4.3).

4.2.1 The CARTwheels algorithm

The first developed approach for redescription mining, CARTwheels [3], works by building two decision trees of predefined depth in the opposite directions, so that they are joined in the leaves (see Figure 4.2, center). The algorithm was originally constructed to work with Boolean attributes. Suppose we have a dataset consisting of two Boolean views W_1 with $V_1 = \{X_1, \dots, X_n\}$ and W_2 with $V_2 = \{Y_1, \dots,\ Y_m\}$ and a set of entities $E = \{e_1, \dots e_s\}$. The CARTwheels algorithm works by constructing multi-class classification trees. In the initial step, each entity is assigned a class depending on the first attribute, from the selected Boolean view, with a value *true*. Suppose that for entity e_k, the first such attribute is X_j, then this entity is assigned target class X_j in multi-class classification to construct the decision trees using variables Y. For all consequent trees, entity membership in the leaves of the opposite tree is used to assign classes for the construction of the next tree. For instance, to construct the bottom tree depicted in Figure 4.2 (middle), one class is used for each leaf of the top tree. Thus, classes $L_1,\ L_2,\ \dots,\ L_8$ correspond to the leaves of the top tree. Each entity $e_k \in L_j$ is assigned to class L_j.

The algorithm uses alternations (see Figure 4.2 center and right)—building trees in alternating fashion (first top tree, then bottom tree, followed by newly constructed top tree etc.). In alternations, trees are regrown to better match the opposing tree. The algorithm performs alternations for a given, predefined number of steps producing redescriptions in the process. Redescriptions are obtained in each alternation step by examining the paths corresponding to the same class from the root node of one tree to the root node of the other tree. Each constructed path represents one redescription.

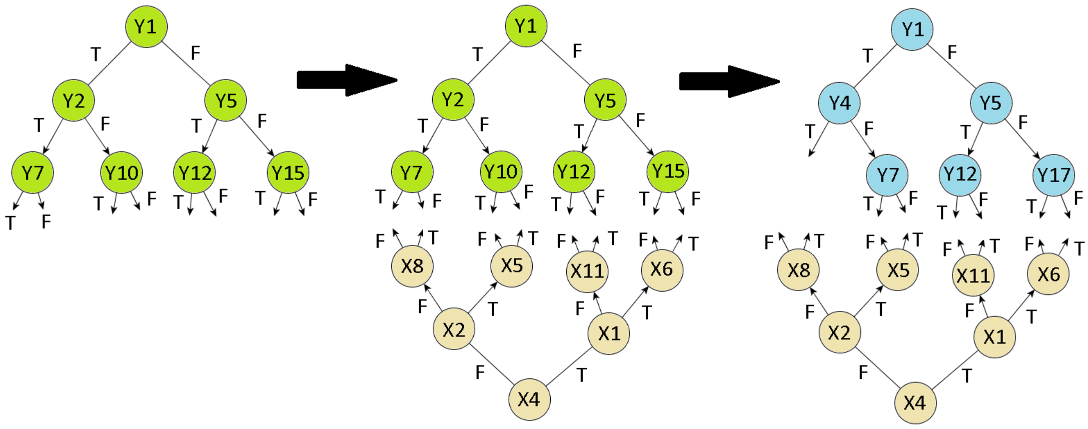


Figure 4.2: Alternations in the CARTwheels algorithm. The initial tree (left) is built by using target classes corresponding to the first attribute contained in the opposite view for which the entity has a value true. Consequent trees use entity membership in the leaves of the opposite tree as targets for tree construction. Classes to construct the bottom tree (middle) are constructed from entity memberships in the leaves of the top tree. The next step is regrowing the top tree to match the bottom tree (right). The process is repeated for a predefined number of iterations.

CARTwheels is a heuristic algorithm that does not perform exhaustive redescription search. It is the first algorithm to utilise decision trees to create redescriptions and it introduced the process of alternations. Its main drawback is that it builds trees of equal (predefined) size at each alternation step but uses only the leafs of the trees to guide the search. As all other existing tree-based redescription mining algorithms, the algorithm has quadratic time complexity with respect to number of entities contained in the dataset but linear with respect to number of attributes. This is in contrast to greedy approaches that have linear complexity with respect to number of entities but potentially quadratic with respect to number of attributes.

4.2.2 The Split trees algorithm

The Split trees algorithm [14], [38] works by alternating decision trees of different (increasing) depth. When both trees reach a predefined maximally allowed depth, they are matched in their leaves, as in the CARTwheels algorithm.

The construction process in the Split trees algorithm is illustrated in Figure 4.3.

The tree of depth one is built by using attributes from V_1. The original implementation, presented in [14], requires the initialization to be performed on the view containing Boolean attributes. These Boolean attributes are chosen in turn to construct targets for the initial decision tree. All entities containing value true for a given attribute are assigned to class C_1 and all other entities to class C_2. Nominal attributes can be transformed to a set of Boolean attributes by creating one Boolean attribute for each categorical value of a nominal attribute. Numerical attributes can be transformed to a set of Boolean attributes by applying attribute discretization techniques. Discretization technique used in [14] includes

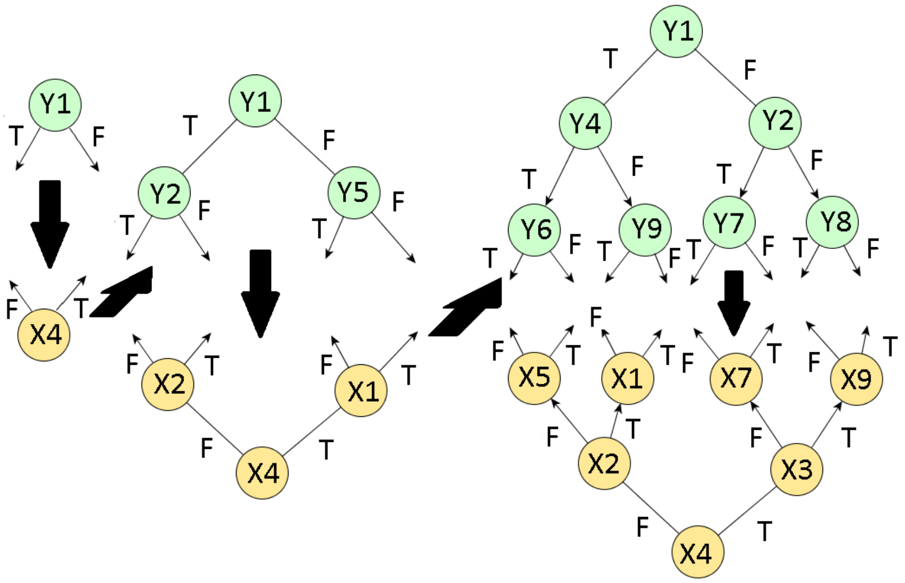


Figure 4.3: Iterations in the Split trees algorithm. The algorithm builds decision trees of different (increasing) depths, at each step until the trees reach a predefined, maximal depth. Trees of maximal depth are matched in their leaves and used to construct redescriptions.

clustering the selected numerical attribute and using information about entity membership in the produced clusters to construct a set of Boolean attributes. The version, presented in the tool Siren [15], works with numerical data but not with missing values.

After the initialization step is performed, the algorithm performs alternations. Each consecutive tree is built based on classes obtained from entity assignments in the leaves of the previously constructed trees (arrows in Figure 4.3). Once both trees are built to the predefined depth, the trees are joined in the leaves and redescriptions are constructed by computing the paths from the root node of one tree to the root node of the second tree (Figure 4.3, right).

The Split trees algorithm is a heuristic algorithm that does not perform exhaustive redescription search. It modifies the alternation process of the CARTwheels algorithm building trees of increasing depth at each step. This allows iteratively refining the subsets of entities obtained using attributes from both views, ultimately leading to increased accuracy of redescriptions. The main drawback of this approach is that parts of the constructed trees are discarded in alternations.

4.2.3 The Layered trees algorithm

The Layered trees algorithm [14], [38] alternates by constructing trees in a layered fashion. Initially, trees of depth one are built (Figure 4.4, left). Next, for each leaf of the initial trees, a tree of depth one is built and appended to the leaves of the initial trees (Figure 4.4, center).

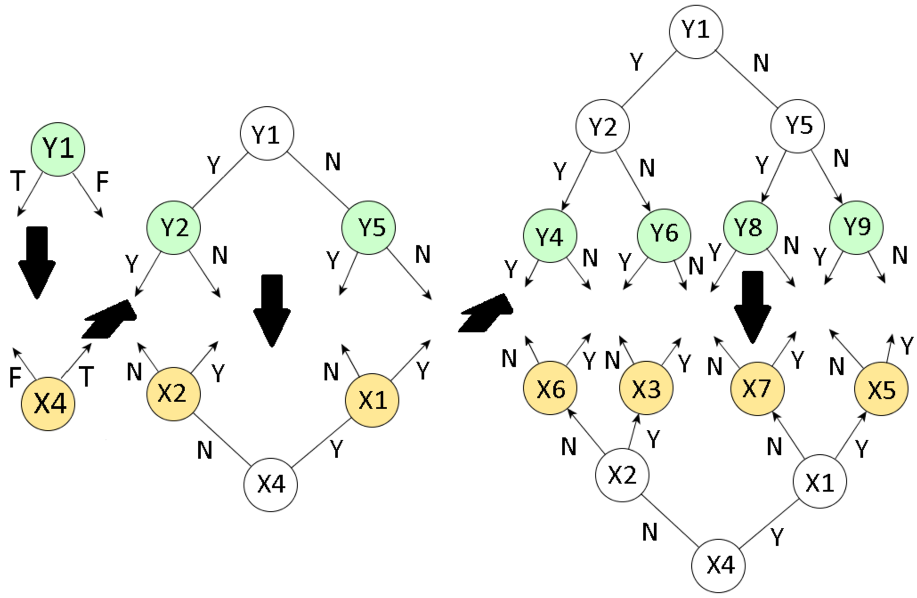


Figure 4.4: Iterations in the Layered trees algorithm. The algorithm builds decision trees of depth one for each leaf of the tree constructed in the previous step or the initial step. The process is repeated until trees reach a predefined, maximal depth. Trees of maximal depth are matched in their leaves and used to construct redescriptions.

The process continues until the trees reach a predefined maximal depth (Figure 4.4, right). The target classes are constructed in the same way as in the Split trees algorithm. As in the Split trees algorithm, redescriptions are constructed by examining the paths corresponding to the same class from the root node of one tree to the root node of the other.

The Layered trees algorithm is a heuristic algorithm that does not perform exhaustive redescription search. It modifies the alternation process of the CARTwheels algorithm building trees of depth one at each alternation step. This allows iteratively refining the subsets of entities obtained using attributes from both views without discarding any constructed nodes. This process ultimately leads to increased accuracy of redescriptions. Since no nodes are discarded and trees are not re-grown as in Split trees, it may use suboptimal splits for redescription construction.

4.3 CLUS-RM: Generating Redescriptions Using Predictive Clustering Trees

The CLUS-RM redescription mining algorithm (presented in [19]) differs from other tree-based approaches in the type of trees used, initialization procedure and tree construction in algorithm alternations which results in different properties of produced redescriptions. It uses multi-target or multi-label predictive clustering trees (PCTs) [17], [35] to create a cluster hierarchy (where each cluster is described by a rule). These rules constitute redescription queries which build redescriptions. Produced rules are used as targets in multi-label classification or multi-target regression setting to find matching queries from the opposing view. Due to the use of multi-target classification and regression properties of PCTs, CLUS-RM can utilize information about all nodes in a constructed PCT to create redescriptions. Because of this and due to the property of inductive transfer [77], it is capable of producing a high number of diverse, accurate redescriptions. Related target variables can carry information about each other. The inductive transfer occurs when information contained within one target variable, about other target variables, are used to improve prediction of those target variables.

Generative grammar of the query language of redescriptions generated by the CLUS-RM algorithm is defined as:

- <literal> → <feature>
- <literal> → ¬ <feature>
- <query> → <literal>
- <query> → $\wedge_{i=1}^{n}$ (<literal>)$_i$, $n \in \mathbb{N}$
- <query> → (<query>) ∨ <literal>
- <query> → ¬ ($\wedge_{i=1}^{n}$ (<literal>)$_i$), $n \in \mathbb{N}$
- <query> → (<query>) ∨ ($\wedge_{i=1}^{n}$ (<literal>)$_i$, $n \in \mathbb{N}$

It is important to notice that the resulting formulas have a form very close to the Disjunctive Normal Form [78]. Transformation of such queries to the Disjunctive Normal Form includes transforming every occurrence of sub-formulas of the form ¬ ($\wedge_{i=1}^{n}$ (<literal>)$_i$, $n \in \mathbb{N}$) to ($\vee_{i=1}^{n}$ ¬(<literal>)$_i$, $n \in \mathbb{N}$).

As any other tree-based redescription mining algorithm, CLUS-RM has many parameters, some of which may be hard to assess a-priori. Some of these parameters are maximal allowed PCT depth, number of iterations and potentially number of restarts (different runs with new random initialization). These parameters are set based on characteristics of desired output and available computational resources. Currently, CLUS-RM uses disjunction operator exclusively to combine queries produced by PCTs. The accuracy and number of redescriptions containing disjunction operator can potentially be increased by using some query-modification procedure similar to greedy atomic updates.

4.4 CLUS-RM Augmented with Random Forest of Predictive Clustering Trees

The CLUS-RM algorithm has been extended [18] to include a random forest of PCTs to create queries used in redescription construction. One PCT model is used to guide the search (nodes from this tree are used as targets in query construction) while the random forest of PCTs is applied to the same targets to create additional queries. As a result, the number of produced redescriptions, their accuracy and diversity increases. Random forest models have several important parameters, the number of trees in a forest and the size of a random subspace of attributes to be used for a single tree construction. The size of a random subspace of attributes is usually set to $\sqrt{n}$ or $log_2(n)$, where n denotes the number of attributes.

However, the diversity of used attributes is very important to produce different redescriptions. Because of this, we compute the size of the random subspace of attributes used in the random forest algorithm so that each attribute occurs in at least one subspace during tree construction with probability p. With the assumption that each attribute contained in a random subspace of size k can be chosen to create a test that splits the entities, the probability of a given attribute to be chosen in a random subspace equals $p_s = \frac{k}{|W_i|}$, for a given view W_i. With the assumption of a fixed tree of depth $d \in \mathbb{N}$, the possible number of tests is $2^d - 1$. The probability that a given attribute does not occur in any test in a given tree is $p_{no} = (1 - \frac{k}{|W_i|})^{(2^d-1)}$. Thus, the probability of an attribute occurring in at least one test equals $p_{aone} = 1-((1-\frac{k}{|W_i|})^{(2^d-1)})$. We denote $q = 2^d-1$. From this, we can compute: $1-p_{aone} = (1-\frac{k}{|W_i|})^q$ which means $1-\sqrt[q]{1-p} = \frac{k}{|W_i|}$. Finally, random subspace size equals $k = |W_i| \cdot (1-\sqrt[q]{1-p})$. Notice that for a small number of attributes $|W_i|$ the size k quickly drops to 0. Because of this, the size of the random subspace is computed as $k = max(\lceil |W_i| \cdot (1-\sqrt[q]{1-p})\rceil,\ \lceil log_2(|W_i|)\rceil)$. Setting higher probability of attribute occurrence increases the size of a random subspace used in random forest construction, effectively increasing the potential for creation of diverse redescriptions.

4.5 Constraint-Based Redescription Mining

Constraint-based redescription mining [11] offers the ability to incorporate expert or domain knowledge in a redescription mining process to produce a selected subset of redescriptions with some predefined properties. Constraints introduced in [11] allow focusing redescription mining on a subset of entities that must be described by the produced redescriptions or a subset of attributes that must occur in redescription queries.

We build upon this work to allow constraint-based redescription mining on numerical and categorical attributes and define several modes of targeted redescription mining. Different modes allow setting constraints of different intensity reflecting the expert's certainty in the relevance and importance of the imposed conditions. Our work on constraint-based redescription mining is presented in [22].

4.6 Techniques for Improving Redescription Quality

As previously mentioned, the CLUS-RM algorithm is capable of producing a high number of diverse redescriptions. Some of these redescriptions can have large queries containing redundancies in attributes or lower accuracy than desired or possible.

We have developed two techniques for improving redescription quality: a) conjunctive refinement procedure aimed at increasing redescription accuracy and b) redescription query size reduction technique aimed at reducing the size of redescription queries without changing the redescription accuracy.

The conjunctive refinement procedure, introduced in [20] uses two existing redescriptions $R_1 = (q_{1,1}, q_{1,2})$ and $R_2 = (q_{2,1}, q_{2,2})$, such that $supp(R_1) \subseteq supp(R_2)$ to produce a new redescription $R_3 = (q_{1,1} \wedge q_{2,1}, q_{1,2} \wedge q_{2,2})$, such that $supp(R_3) = supp(R_1)$. We have proven that $J(R_3) \geq J(R_1)$. A direct side effect of using this procedure is the increase in the number of attributes in queries of newly produced redescriptions.

The redescription query size reduction procedure [19] aims at reducing the number of attributes contained in redescription queries without changing the redescription support or accuracy. Redundancy in attributes can occur as a natural consequence of PCT construction or during the application of the conjunctive refinement procedure. One step in the redescription query size reduction is equivalent to solving the set cover problem (SCP) [79], which is known to be NP-complete in its decision variant. Because of this, the procedure used is a heuristic method that does not guarantee that the minimal possible number of attributes, describing some subset of entities, will be found. Any algorithm for solving a set covering problem [80] can be used to reduce the size of redescription queries in the setting defined in [19].

4.7 Related Publications

Details of the redescription mining algorithm CLUS-RM and the redescription query minimization procedure are described in the following publication (included in this Chapter):

M. Mihelčić, S. Džeroski, N. Lavrač, and T. Šmuc, "Redescription mining with multi-target predictive clustering trees," in *Proceedings of the 4th International Workshop, New Frontiers in Mining Complex Patterns, NFMCP 2015, Held in conjunction with ECML-PKDD 2015, Porto, Portugal, September 7, 2015, Revised Selected Papers*, M. Ceci, C. Loglisci, G. Manco, E. Masciari, and Z. W. Ras, Eds. Cham: Springer International Publishing, 2016, pp. 125–143.

The author contributions are as follows. Matej Mihelčić constructed and implemented the CLUS-RM algorithm based on multi-label classification and multi-target regression PCTs. He also introduced the concept of redescription set optimization by redescription exchange, implemented the required methodology, constructed and implemented several new redescription evaluation quality measures. He constructed and implemented the methodology for the redescription query size reduction, computed the computational time complexity of the algorithm, performed all the experiments and analysed the results. He wrote the majority of the manuscript and actively participated in writing all manuscript revisions. Nada Lavrač suggested performing the research in the field of redescription mining. The idea from Sašo Džeroski to use Predictive Clustering Trees to produce descriptive rules on multi-view data led to the development of the PCT-based redescription mining algorithm. Tomislav Šmuc initiated the idea of using rules as targets in algorithm alternations and suggested using the currently implemented initialization procedure. Tomislav Šmuc, Sašo Džeroski and Nada Lavrač participated in writing, proofreading and correcting the manuscript text.

Redescription mining with multi-target Predictive Clustering Trees

Matej Mihelčić[1,3], Sašo Džeroski[2,3], Nada Lavrač[2,3], and Tomislav Šmuc[1]

[1] Ruđer Bošković Institute
Bijenička cesta 54, 10000 Zagreb, Croatia
{matej.mihelcic, tomislav.smuc}@irb.hr
[2] Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
{saso.dzeroski, nada.lavrac}@ijs.si
[3] Jožef Stefan International Postgraduate School
Jamova cesta 39, 1000 Ljubljana, Slovenia

Abstract. Redescription mining is a field of knowledge discovery that aims to find different descriptions of subsets of elements in the data by using two or more disjoint sets of descriptive attributes. The ability to find connections between different sets of descriptive attributes and provide a more comprehensive set of rules makes it very useful in practice. In this work, we introduce redescription mining algorithm for generating and iteratively improving a redescription set of user defined size based on multi-target Predictive Clustering Trees. This approach uses information about element membership in different generated rules to search for new redescriptions and is able to produce highly accurate, statistically significant redescriptions described by Boolean, nominal or numeric attributes. As opposed to current tree-based approaches that use multi-class or binary classification, we explore benefits of using multi target classification and regression to create redescriptions. The process of iterative redescription set improvement is illustrated on the dataset describing 199 world countries and their trading patterns. The performance of the algorithm is compared against the state of the art redescription mining algorithms.

Keywords: knowledge discovery, redescription mining, predictive clustering trees, world countries

This is a post-peer-review, pre-copy edit version of an article published in Post-proceedings of the NFMCP15 workshop. The final authenticated version is available online at: https://doi.org/10.1007/978-3-319-39315-5_9.

1 Introduction

Pattern mining [1, 13] aims at discovering descriptive rules learned from data. Redescription mining [19] shares this goal but tries to find different descriptions of patterns by using two or more disjoint sets of descriptive attributes which are finally presented to the user. It is an unsupervised, descriptive knowledge discovery task. This analysis allows finding similarities between different elements and connections between different descriptive attribute sets (views) which ultimately lead to better understanding of the underlying data. Redescription mining is highly applicable in biology, economy, pharmacy, ecology and many other fields, where it is important to understand connections between different descriptors and to find regularities that are valid for different element subsets. Redescriptions are represented in the form of rules and the aim is to make these rules understandable and interpretable.

The field of redescription mining was introduced by Ramakrishnan et al. [19]. Their paper presents a novel algorithm to mine redescriptions based on decision trees, called the CARTwheels. The algorithm works by building two decision trees (one for each view) that are joined in the leaves. Redescriptions are found by examining the paths from the root node of the first tree to the root node of the second and the algorithm uses multi class classification to guide the search between the two views. Other approaches to mine redescriptions include approach proposed by Zaki and Ramakrishnan [23] which uses a lattice of closed descriptor sets to find redescriptions. Further, Parida and Ramakrishnan [17] introduce algorithms for mining exact and approximate redescriptions, Gallo et al. [10] present the greedy and the MID algorithm based on frequent itemset mining.

Galbrun and Miettinen [6] present a novel greedy algorithm for mining redescriptions. In this work they extend the greedy approach by Gallo et al. [10] to work on numeric data since all previous approaches worked only on Boolean data. Redescription mining was extended by Galbrun and Kimming to a relational [5] and by Galbrun and Miettinen to the interactive setting [8]. Recently, two novel tree-based algorithms were proposed by Zinchenko [24], which explore using decision trees in a non-Boolean setting and present different methods of layer by layer tree construction, which allows making informed splits based on nodes at each level of the tree.

In this work, we explore creation and iterative improvement of redescription sets containing a user defined number of redescriptions. With this goal in mind, we developed a novel algorithm for mining redescriptions based on multi-target predictive clustering trees (PCTs) [3, 14]. Our approach uses multi-target classification or regression to find highly accurate, statistically significant redescriptions, which differentiates it from other tree based approaches, especially the CARTwheels approach. Each node in a tree represents a separate rule that is used as a target in the construction of a PCT from the opposite view. Using multi-target PCTs allows us to build one model to find multiple redescriptions using nodes at all levels of the tree, further it allows to find features that are connected with multiple target features (rules) and finally due to inductive transfer [18], multi-target trees can outperform single label classification or regression trees. We have developed a procedure for rule minimization that allows us to find the smallest subset of attributes that describe a given pattern, thus we have the ability to get shorter rules even when using trees of bigger depth size. The approach is related to multi-view [2] and multilayer [11] clustering, though the main goal here is to find accurate redescriptions of interesting subsets of data, while clustering tends to find clusters that are not always easy to interpret.

After introducing the necessary notation (Section 2), we present the algorithm, introduce the procedure for rule minimization and perform the run-time analysis of redescription mining process (Section 3). We use the algorithm to iteratively improve redescription set describing 199 different world countries based on their trading behaviour [21] and general country information [22] for the year 2012 (Section 4). The main focus is on rules containing only logical conjunction operators, since these rules are the most interpretable and very easy to understand. In Section 5 we analyse redescription sets mined with one state of the art redescription mining algorithm, optimize redescription sets of equal size with our approach, compare these sets by using several criteria and discuss the results. Finally, we conclude and outline directions for future work in Section 6.

2 Notation and definitions

Redescription mining in general considers redescriptions constructed on a set of views $\{W_1, W_2, \ldots, W_n\}$, however in this paper we use only two views $\{W_1, W_2\}$. The corresponding attribute (variable) sets are denoted by V_1 and V_2. Each view contains the same set of $|E|$ elements and two different sets of attributes of size $|V_1|$ and $|V_2|$. Value $W_1(i, j)$ is the value of element e_i for the attribute a_j in view W_1. The data $D = (V_1, V_2, E, W_1, W_2)$ is a quintuple of the attribute sets, the element set, and the appropriate view mappings. A query (denoted q) is a logical formula F, where q_1 contains literals from V_1. The set of elements described by a query is called its support. A redescription $R = (q_1, q_2)$ is defined as a pair of queries, one for each view in the data. The support of a redescription is the set of elements supported by both queries that constitute this redescription: $supp(R) = supp(q_1) \cap supp(q_2)$. We use $attr(R)$ to denote the multiset of attributes used in the redescription R. The accuracy of a redescription $R = (q_1, q_2)$ is measured using the Jaccard coefficient (Jaccard similarity index):

$$JS(R) = \frac{|supp(q_1) \cap supp(q_2))|}{|supp(q_1) \cup supp(q_2)|}$$

The Jaccard coefficient is not the only measure used in the field because it is possible to obtain redescriptions covering huge element subsets that necessarily have very good overlap of their queries. In this cases it is preferred to have redescriptions that reveal some more specific knowledge about the studied problem that is harder to obtain by random sampling from the underlying data distribution. This is why we compute the statistical significance (p-value) of each obtained redescription. We denote the marginal probability of a query q_1, q_2 with $p_1 = \frac{supp(q_1)}{|E|}$ and $p_2 = \frac{supp(q_2)}{|E|}$ respectively. We define the set of elements in the intersection of the queries with $o = supp(q_1) \cap supp(q_2)$. The corresponding p-value [9] is defined as

$$pV(q_1, q_2) = \sum_{n=|o|}^{|E|} \binom{|E|}{n} (p_1 \cdot p_2)^n \cdot (1 - p_1 \cdot p_2)^{|E|-n}$$

The p-value tells us if we can dismiss the null hypothesis that assumes that we obtained a given subset of elements by joining two random rules with marginal probabilities equal to the fraction of covered elements. If the obtained p-value is lower than some predefined threshold, called the significance level, then this null hypothesis should be rejected. This is a somewhat optimistic criterion, since the assumption that all elements can be sampled with equal probability need not hold for all datasets.

3 The CLUS-RM algorithm

In this section, we describe the algorithm for mining redescriptions named CLUS-RM, that at each step improves the redescription set of the size defined by the user. The algorithm uses multi-target predictive clustering trees (PCTs) [3, 14] to create a cluster hierarchy that is later transformed into redescriptions. We start by explaining the pseudo code of the algorithm (Algorithm 1) and then go into the details of each procedure in the algorithm.

Algorithm 1 The CLUS-RM algorithm

```
Input: First view data ($W_1$), Second view data ($W_2$), Settings file
Output: A set of redescriptions $\mathcal{R}$
 1: procedure CLUS-RM
 2:     [$\mathcal{D}_{W_1init}$, $\mathcal{D}_{W_2init}$] ← prepareTargetsForInitialPCT($W_1$,$W_2$)
 3:     [PCT$W_1$, PCT$W_2$] ← createSidesInitialPCT($\mathcal{D}_{W_1init}$, $\mathcal{D}_{W_2init}$)
 4:     [R$W_1$, R$W_2$] ← extractRules(PCT$W_1$, PCT$W_2$)
 5:     initializeArrays(elFreq, attrFreq, redScoreEl, redScoreAt, numEx, numAttr,
                         numRetRed)
 6:     while RunInd<maxIter do
 7:         [TmpR$W_1$, TmpR$W_2$] ← emptyRuleSet()
 8:         [$\mathcal{D}_{W_1Targ}$, $\mathcal{D}_{W_2Targ}$] ← prepareTargets(R$W_2$, R$W_1$)
 9:         [PCT$W_1$, PCT$W_2$] ← createPCT($\mathcal{D}_{W_1Targ}$, $\mathcal{D}_{W_2Targ}$)
10:         TmpR$W_1$ ← TmpR$W_1$ $\cup_*$ extractRules(PCT$W_1$)
11:         TmpR$W_2$ ← TmpR$W_2$ $\cup_*$ extractRules(PCT$W_2$)
12:         R$W_1$ ← R$W_1$ ∪ TmpR$W_1$
13:         R$W_2$ ← R$W_2$ ∪ TmpR$W_2$
14:         $\mathcal{R}$ ← MineRed(R$W_1$, R$W_2$, expansionType,
                ConstSet, iteration, opSet, elFreq, attrFreq, redScoreEl, redScoreAt)
15:     return $\mathcal{R}$
```

The algorithm starts by creating initial clusters for both views (line 2 and 3 in Algorithm 1) which is achieved by transforming a non-labeled dataset into a labeled dataset of positive elements and artificially generated negative elements.

For each element in the original view, we construct one negative, synthetic element (see Figure 1) in such a way so that the original correlations among the attributes are broken. We achieve this by random shuffling of attribute values between the elements. The procedure allows experimentation with the number of shuffling steps and the number of attributes that are copied from the original elements to the artificial element. Complete randomization is achieved when the number of shuffling steps equals the number of attributes in the dataset and exactly one attribute value is copied to the artificial element at each step from a randomly chosen original element. The original elements are assigned a target label of 1.0, while the artificial elements are assigned a target label of 0.0 (see Table 1). The division between the original and the artificial elements (the idea previously used in [11]), allows us to construct a cluster hierarchy, simultaneously creating descriptions of the original elements. The described procedure is one possible way to construct the initial clusters; other approaches include assigning a random target attribute or using clusters computed by some other clustering algorithm. However, the initialization procedure used in our algorithm should preserve any strong (specific) connections and correlations that exist in the original data which are broken by using an approach that assigns random target labels.

Table 1: Creation of artificial elements for the random initialization procedure.

(a) Original dataset for view 1

Entity	W_1A_1	W_1A_2	W_1A_3
E_1	1.1	2.5	3.4
E_2	1.5	2.2	4.0
E_3	5.5	-0.6	-0.2
E_4	4.4	-0.2	2.0
E_5	3.2	1.7	2.9

(b) Original dataset for view 2

Entity	W_2A_1	W_2A_2	W_2A_3
E_1	TRUE	FALSE	FALSE
E_2	TRUE	TRUE	FALSE
E_3	FALSE	FALSE	TRUE
E_4	TRUE	TRUE	TRUE
E_5	TRUE	FALSE	TRUE

(c) Initial dataset for view 1

Entity	W_1A_1	W_1A_2	W_1A_3	Target
E_1	1.1	2.5	3.4	1.0
E_2	1.5	2.2	4.0	1.0
E_3	5.5	-0.6	-0.2	1.0
E_4	4.4	-0.2	2.0	1.0
E_5	3.2	1.7	2.9	1.0
E_1'	4.4	2.5	2.9	0.0
E_2'	3.2	-0.6	4.0	0.0
E_3'	3.2	-0.6	2.9	0.0
E_4'	4.4	-0.2	4.0	0.0
E_5'	5.5	1.7	2.9	0.0

(d) Initial dataset for view 2

Entity	W_2A_1	W_2A_2	W_2A_3	Target
E_1	TRUE	FALSE	FALSE	1.0
E_2	TRUE	TRUE	FALSE	1.0
E_3	FALSE	FALSE	TRUE	1.0
E_4	TRUE	TRUE	TRUE	1.0
E_5	TRUE	FALSE	TRUE	1.0
E_1'	TRUE	FALSE	TRUE	0.0
E_2'	FALSE	FALSE	TRUE	0.0
E_3'	TRUE	TRUE	TRUE	0.0
E_4'	FALSE	TRUE	FALSE	0.0
E_5'	FALSE	FALSE	TRUE	0.0

After creating the initial dataset, we build predictive clustering trees on both views by performing regression on the target label and using other attributes as descriptive. The decision to use regression trees instead of decision trees is purely technical, since it generates more rules because of the additional threshold associated with the target variable. These trees are converted to rules that describe element sets and are necessary for the next step of the algorithm. The rule lists RW_1 and RW_2 contain generated rules, and a new rule is added to the list if it differs from all other rules in a predefined number of attributes or if it describes a new unique element subset (the $\cup_*$ operator in Algorithm 1). The iterative process of the algorithm begins right after rule creation. Here, we create targets based on the rules obtained in the previous step or in the initialization step. The Rules obtained by predictive clustering on W_1 are used to build targets for clustering on W_2 (denoted W_1T_1, W_1T_2), and vice versa. For each element in the dataset we assign label 1.0 if the element is described by some specific rule, otherwise 0.0 (see Table 2). For example, the attribute W_2T_1 from dataset for view 1 represents the condition $IF\ W_2A_1 = TRUE$ (constructed on dataset for view 2), which describes elements E_1, E_2, E_4, E_5. By placing this target attribute in the view 1 dataset, we guide the PCT construction to create a cluster containing and describing the same set of elements with descriptive variables of view 1 (a choice that satisfies this condition is $IF\ W_1A_3 > 0$).

Table 2: Intermediate generation of labels based on discovered rules.

(a) Dataset for view 1

E	W_1A_1	W_1A_2	W_1A_3	W_2T_1	W_2T_2
E_1	1.1	2.5	3.4	1.0	0.0
E_2	1.5	2.2	4.0	1.0	0.0
E_3	5.5	-0.6	-0.2	0.0	0.0
E_4	4.4	-0.2	2.0	1.0	0.0
E_5	3.2	1.7	2.9	1.0	1.0

(b) Dataset for view 2

E	W_2A_1	W_2A_2	W_2A_3	W_1T_1	W_1T_2
E_1	TRUE	FALSE	FALSE	0.0	1.0
E_2	TRUE	TRUE	FALSE	0.0	1.0
E_3	FALSE	FALSE	TRUE	1.0	0.0
E_4	TRUE	TRUE	TRUE	1.0	0.0
E_5	TRUE	FALSE	TRUE	1.0	1.0

Rules obtained in the previous step are combined into redescriptions if they satisfy a given set of constraints $ConstSet$. The set of constraints consists of minimal Jaccard coefficient ($minJS$), maximum allowed p-value ($maxPval$) and minimum and maximum support ($minSupp$, $maxSupp$) which have to be satisfied for a redescription to be considered as a candidate for the redescription set.

3.1 The procedure for creating redescriptions

The algorithm for creating redescriptions from rules (Algorithm 2) joins view 1 rules (or its negation, if allowed by the user) with rules (or its negation) from view 2 (see Figure 1 and line 2 in Algorithm 2). We distinguish three cases of creating redescriptions from rules (expansion types):

1. Unguided initial: $UInit \leftarrow (RW_1 \times^{opSet\setminus\{\vee\}}_{ConstSet} RW_2)$
2. Unguided: $U \leftarrow (RW_{1_{newRuleIt}} \times^{opSet\setminus\{\vee\}}_{ConstSet} RW_{2_{newRuleIt}})$
3. Guided: $G \leftarrow (RW_{1_{newRuleIt}} \times^{opSet\setminus\{\vee\}}_{ConstSet} RW_{2_{oldRuleIt}}) \cup$ $(RW_{1_{oldRuleIt}} \times^{opSet\setminus\{\vee\}}_{ConstSet} RW_{2_{newRuleIt}})$

The $\times^{opSet}_{ConstSet}$ operator denotes a Cartesian product of two sets, allowing the use of logical operators from *opSet* and leaving only those redescriptions that satisfy a given set of constraints $ConstSet$. The unguided expansion allows obtaining redescriptions with more diverse subsets of elements that can later be improved through the iteration process.

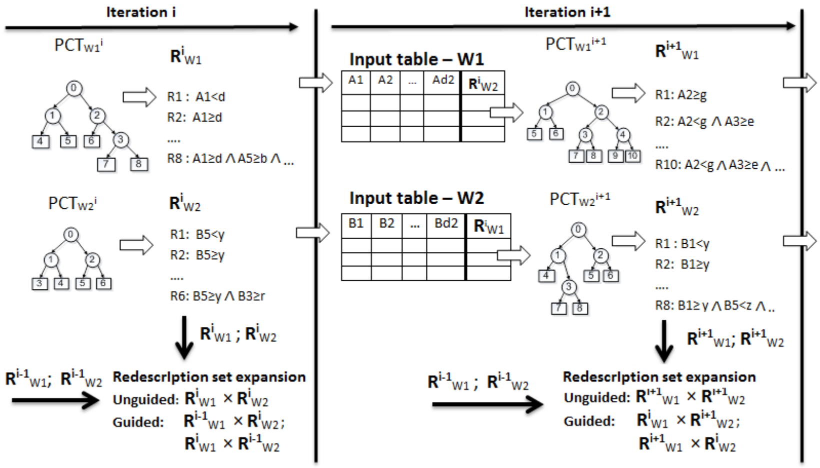

Iteration i
$PCT_{W1}{}^{i}$
R^{i}_{W1}
0
1
2
3
4
5
6
7
8
R1 : A1<d
R2: A1≥d
....
R8 : A1≥d ∧ A5≥b ∧ ...
$PCT_{W2}{}^{i}$
R^{i}_{W2}
R1: B5<y
R2: B5≥y
....
R6: B5≥y ∧ B3≥r
R^{i}_{W1} ; R^{i}_{W2}
R^{i-1}_{W1}; R^{i-1}_{W2}
Redescription set expansion
Unguided: $R^{i}_{W1} \times R^{i}_{W2}$
Guided: $R^{i-1}_{W1} \times R^{i}_{W2}$; $R^{i}_{W1} \times R^{i-1}_{W2}$
Iteration i+1
Input table – W1
A1 A2 ... Ad2 R^{i}_{W2}
Input table – W2
B1 B2 ... Bd2 R^{i}_{W1}
$PCT_{W1}{}^{i+1}$
R^{i+1}_{W1}
R1: A2≥g
R2: A2<g ∧ A3≥e
....
R10: A2<g ∧ A3≥e ∧ ...
9
10
$PCT_{W2}{}^{i+1}$
R^{i+1}_{W2}
R1 : B1<y
R2: B1≥y
....
R8: B1≥y ∧ B5<z ∧ ..
R^{i+1}_{W1}; R^{i+1}_{W2}
R^{i-1}_{W1} ; R^{i-1}_{W2}
Redescription set expansion
Unguided: $R^{i+1}_{W1} \times R^{i+1}_{W2}$
Guided: $R^{i}_{W1} \times R^{i+1}_{W2}$; $R^{i+1}_{W1} \times R^{i}_{W2}$

Fig. 1: Illustration of rule, redescription construction and iterations

The algorithm finds first *numRed* redescriptions and then iteratively enriches this set by exchanging the redescription with the worst comparative score with the newly created redescription (lines 3-14 in Algorithm 2). The algorithm uses 4 arrays ($elFreq$, $attrFreq$, $redScoreEl$, $redScoreAt$) to incrementally add and improve redescriptions in the redescription set. The element/attribute frequency arrays contain the number of times each element/attribute from the dataset occurs in redescriptions from a redescription set. Redescription scores are computed as $redScoreEl(R) = \sum_{e\in supp(R)}(elFreq[e]-1)$, and $redScoreAt(R) = \sum_{a\in attr(R)}(attrFreq[a]-1)$. The score of a new redescription is computed in the same way by using existing frequencies from the set. If the algorithm finds a redescription R' such that $R_i = argmax_{R\in\mathcal{R}|\ R.pval\geq R'.pval}score(R', R)$, where $score(R', R) = (\frac{(1.0-R'.elSc+1.0-R'.atrSc+R'.JS}{3} - \frac{(1.0-R.elSc+1.0-R.attrSc+R.JS)}{3})$, all arrays are updated so that the frequencies of elements described by R_i and attributes contained in it's queries are decreased by one, while the frequencies of elements and attributes associated with R' are increased. This score favours redescriptions that describe elements with low frequency by using non frequent attributes. At the same time it finds as accurate and significant redescriptions as possible.

Algorithm 2 MineRed

```
Input: $RW_1$, $RW_2$, expansion type, ConstSet, iteration number, opSet, elFreq, attrFreq, redScoreEl, redScoreAt
Output: A set of redescriptions $\mathcal{R}$
 1: procedure MineRed
 2:     expansionSet ← returnExpansionSet(expansionType, opSet, $RW_1$, $RW_2$)
 3:     for $R' \in expansionSet$ do
 4:         if $|\mathcal{R}|<$ConstSet.MaxRed then
 5:             updateFrequencies(elFreq, attrFreq, R')
 6:             $\mathcal{R} \leftarrow \mathcal{R} \cup R'$
 7:             if $|\mathcal{R}| ==$ ConstSet.MaxRed then
 8:                 for $R \in \mathcal{R}$ do
 9:                     computeScores(elFreq,attrFreq, redScoreEl, redScoreAt, R)
10:         else if $|\mathcal{R}| ==$ ConstSet.MaxRed then
11:             compScore(elFreq,attrFreq, redScoreEl, redScoreAt, R')
12:             $R_b \leftarrow argmax_{R\in\mathcal{R}|\ R.pval\geq R'.pval}\ score(R', R)$
13:             updtFreqAndScores(elFreq, attrFreq, redScoreEl, redScoreAt, R', R)
14:             $\mathcal{R} \leftarrow \mathcal{R}\backslash R_b \cup R'$
15:     if $\vee \in$ opSet then
16:         for R$\in \mathcal{R}$ do
17:             if expansionType==unguidedExpansion AND iteration==0 then
18:                 $ind \leftarrow 0$
19:             else
20:                 $ind \leftarrow newRuleIt$
21:             $r'_{W_1} \leftarrow argmax(R.maxRef(r), R.maxRef(\neg r),\ r \in RW1_{ind})$
22:             $R_{ref} \leftarrow (r'_{W_1} \vee R.rW_1 \times R.rW_2)$
23:             $r'_{W_2} \leftarrow argmax(R_{ref}.maxRef(r), R_{ref}.maxRef(\neg r),\ r \in RW2_{ind})$
24:             $R_{ref} \leftarrow (R_{ref}.rW_1 \times r'_{W_2} \vee R.rW_2)$
25:             updtFreqAndScores(elFreq, attrFreq, redScoreEl, redScoreAt, R, $R_{ref}$)
26:             $\mathcal{R} \leftarrow \mathcal{R}\backslash R \cup R_{ref}$
27:     return $\mathcal{R}$
```

Element weighting has been used before in subgroup discovery [12, 15] to model covering importance for elements. Our approach is similar but uses different weighting mechanism, adapts it to the redescription mining setting by combining element and attribute weights and incorporates it into the framework of iterative redescription set refinement in which some redescriptions can be replaced with more suitable candidates.

The algorithm can use three types of logical operators (disjunction, conjunction and negation). The disjunction operator is used to increase redescription accuracy and support (lines 15-26 in Algorithm 2). For a redescription $R = (q_1, q_2)$, we find rules r that maximize:

1. $JS(supp(q_1 \vee r)\backslash supp(R), supp(q_2)\backslash supp(R))$
2. $JS(supp(q_1 \vee \neg r)\backslash supp(R), supp(q_2)\backslash supp(R))$
3. $JS(supp(q_1)\backslash supp(R), supp(q_2 \vee r)\backslash supp(R))$
4. $JS(supp(q_1)\backslash supp(R), supp(q_2 \vee \neg r)\backslash supp(R))$

The rule r is found so that it covers elements that are supported by q_2 but not by q_1 $(R.maxRef(r'),\ r' \in RW_1)$ and vice versa.

3.2 Rule size minimization

Rule minimization procedure is applied in the final step of redescription set creation. The main goal of this procedure is to find a minimal attribute set for all rules contained in redescriptions that describe the same pattern as the original redescription. This leads to better understandability and readability of returned redescriptions.

The method minimizes conjunctive formulas $F = v_1 \wedge v_2 \wedge v_3 \wedge v_4 \wedge \cdots \wedge v_n$, where each v_i denotes one literal of the form $v_i = c$ in the case of Boolean or categorical attributes or $c_1 \leq v_i \leq c_2$ in the case of numerical attributes. The procedure chooses each v_i in turn, computes $\mathcal{S}_{v_i} = supp(v_i) \backslash supp(F)$ and then finds the minimal set $\mathcal{T} = \{v_k, \ldots, v_m\}$ such that $\forall e \in \mathcal{S}_{v_i},\ \exists v_j \in \mathcal{T},\ e \notin supp(v_j)$ and $\cap_k v_k = supp(F),\ v_k \in \mathcal{T}$ (see Figure 2). The procedure returns a family of sets $\mathcal{F} = \{\mathcal{T}_i,\ i = 1, \ldots, n\}$ and chooses the representative set containing the smallest number of attributes.

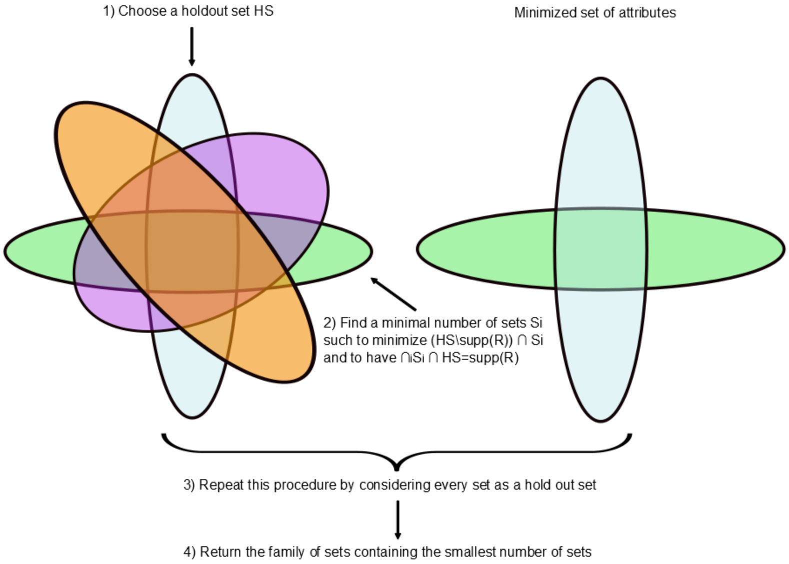

1) Choose a holdout set HS
Minimized set of attributes
2) Find a minimal number of sets Si such to minimize (HS\supp(R)) ∩ Si and to have ∩iSi ∩ HS=supp(R)
3) Repeat this procedure by considering every set as a hold out set
4) Return the family of sets containing the smallest number of sets

Fig. 2: Rule minimization procedure

The procedure is related to a procedure for finding a minimal set of generators in [23]. It is constructed with a purpose of minimizing rules contained in already constructed redescriptions whereas minimal set of generators is used to construct redescriptions which requires it to compute a closed lattice of descriptors.

3.3 Algorithm time complexity

In this subsection we analyse the algorithm's time complexity. We start from the known results [20] that predictive clustering tree construction has the worst time complexity of $O(z \cdot m \cdot |E|^2)$ to completely induce the tree, where m denotes the number of descriptive variables in a selected view and z the total number of internal nodes in the tree.

We use the HashSet and the HashMap data structure with open addressing to store elements which have the time complexity of $O(1)$ for add, remove, contains and size assuming the hash function behaves in a random enough manner (uniform hashing).

The initialization step has the complexity of $O(|E| \cdot (|V_1| + |V_2|))$ and the PCT to rules transformation has the complexity of $O(z)$. Creation of redescriptions via extraction/filtering of pairs obtained from Cartesian product of two rule sets has the worst time complexity of $O(n + n')$, where n equals the number of elements covered by the rule created on $W1$ and n' denotes the number of elements covered by the rule created on $W2$. To compute the Cartesian product of two rule sets we make $\sum_{i \in R_L} \sum_{j \in R_R} (n_i + n_j)$ steps. As both $n \leq |E|$ and $n' \leq |E|$, the worst time complexity of this step is $O(z^2 \cdot |E|)$. However, if we have a balanced tree, the complexity is closer to $O(z \cdot d \cdot |E|)$, where d equals the tree depth. Updating the attribute and element frequency tables and the total redescription scores has the complexity of $O(|E|)$. The computation of rules containing negation and disjunction operators has a complexity of $O(z \cdot |E|)$.

The minimization procedure has the time complexity of $O(|\mathcal{R}| \cdot ((a+a') \cdot |E| + (a^3 + a'^3) \cdot |E|))$, where a, a' represent the number of attributes in redescription rules which are constrained with the tree depth d (or a constant multiple of d in case of rules containing disjunctions). As the the number of elements in support of such constrained attributes is much smaller then $|E|$, the worst case time complexity is $O(d^3 \cdot |E|)$.

The algorithm time complexity is: $O(|E| \cdot (|V_1| + |V_2|) + z \cdot |V_1| \cdot |E|^2 + z \cdot |V_2| \cdot |E|^2 + 2 \cdot z + z^2 \cdot |E| + z^2 \cdot |E| + 2 \cdot z \cdot |E| + d^3 \cdot |E|)$ which is $O(z \cdot (|V_1| + |V_2|) \cdot |E|^2 + z^2 \cdot |E|)$. The pessimistic worst time complexity assuming inadequate hashing function is $O(z \cdot (|V_1| + |V_2|) \cdot |E|^2 + z^2 \cdot |E|^2)$.

Optimizations that could speed up computing redescriptions include the use of rule indexing that would allow combining only those rules certain to cross the user defined thresholds and Local Sensitive Hashing [4].

4 Mining redescriptions on data describing countries

We present the experimental results of mining redescriptions with our algorithm on data describing 199 world countries in the year 2012. ([11, 21, 22]). The dataset has two views, both containing numerical attributes with possible missing values. One view contains 312 attributes representing the importance of import and export of different commodities for countries, while the second view contains 49 attributes with country information provided by the World Bank.

There are several techniques described in [6] for computing Jaccard coefficient when data contains missing values. We compute the Jaccard coefficient guided by the principle that an element can not be in a support of a rule containing only conjunction operator if it has missing values for some of the attributes contained in a condition of a rule. We use notation from [6] to denote $E_{1,1} = supp(q_1) \cap supp(q_2)$, $E_{1,0} = supp(q_1) \backslash supp(q_2)$, $E_{0,1} = supp(q_2) \backslash supp(q_1)$, $E_{1,?} = supp(q_1) \cap missing(q_2)$, $E_{?,1} = missing(q_1) \cap supp(q_2)$, where $R = (q_1, q_2)$ and $missing(q)$ represents a set of elements for which we can not determine if they are in support of q due to missing values. We define the Jaccard coefficient as:

$$JS_m(R) = \frac{|E_{1,1}|}{|E_{1,0}| + |E_{0,1}| + |E_{1,1}| + |E_{1,?}| + |E_{?,1}|}$$

It holds that $JS_{pes}(R) \leq JS_m(R) \leq JS_{opt}(R)$, where JS_{opt} and JS_{pes} denote optimistic and pessimistic estimate of JS when dealing with missing values.

The algorithm was tested with 50, 200, 800 iterations and only rules containing the conjunction operator were allowed. For each number of iterations, we performed 10 runs of the algorithm, computed redescription sets containing 50 redescriptions and measured the average Jaccard coefficient and the average redescription support. Allowed redescription supports were in range $[5, 120]$, the maximum p-value equalled 0.01 and the minimum Jaccard coefficient was 0.6. We used complete randomization in initialization procedure.

Figure 3 shows that with increased number of iterations, the algorithm finds redescriptions with higher accuracy, but describing smaller subsets of countries. The mean value of the total overall coverage of elements in the redescription set varies between 47% and 53%. This indicates that the algorithm managed to find highly accurate redescriptions describing a significant number of total elements from the dataset.

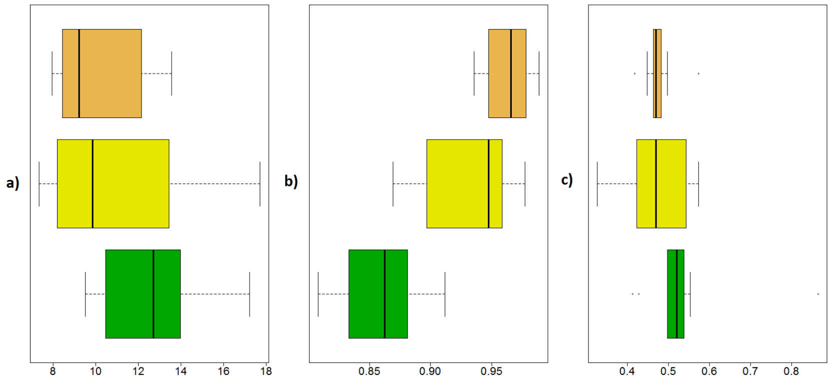


Fig. 3: A summary of the results for different numbers of algorithm runs (top to bottom: 800, 200, 50): average redescription support size a), average Jaccard coefficient b), fraction of all elements described by a redescription c).

We demonstrate one highly accurate, statistically significant redescription mined on the Country dataset. Several additional examples can be seen in [16].

```
W1R: EMPL_BAD >= 5.6 <= 12.5 AND POP_14 >= 13.166 <= 18.2591 AND
     CRED >= 99.2251 <= 305.0869
W2R: E/I_MED_PH >= 0.927 <= 4.563 AND E/I_FB >= 0.381 <= 1.46 AND
     E/I_PULP_WP >= 0.332 <= 859.221
```

This redescription describes 14 world countries (United Kingdom, Switzerland, Sweden, Spain, Singapore, Netherlands, Malta, Luxembourg, Germany, France, Finland, Denmark, Cyprus and Austria) with Jaccard coefficient 1.0. We found that the vulnerable employment of these countries ranges from $[5.6, 12.5]\%$, the percentage of population aged $0-14$ is in $[13.2, 18.3]\%$, and the domestic credit to private sector is in $[99.2, 305.1]\%$ of the GDP. In addition, export to import ratio of medicinal and pharmaceutical products is in $[0.9, 4.6]\%$, export to import ratio of basic food is in $[0.4, 1.5]\%$ and this ratio for pulp and waste paper is in $[0.3, 859.2]\%$. This is a statistically highly significant redescription with a p-value of $1.5 \cdot 10^{-13}$, it contains 3 descriptive variables for view 1 and 3 variables for view 2. It is a medium size redescription, based on its rule size.

5 Algorithm evaluation and comparison

In this section, we compare rules produced by our algorithm with the current state of the art algorithm ReReMi, described in [9]. We used the Siren tool [7] to perform redescription mining with the ReReMi algorithm on the Country dataset described in Section 4. The layered/split tree algorithms (described in [24]) currently do not work with data that contain missing values.

Redescription mining algorithm comparison was mainly done in the literature by selecting and discussing properties of the individual redescriptions. We try to make objective evaluation of redescription sets produced by different algorithms by using the same set of constraints to construct redescriptions. Another condition we imposed is to have the same size of the final redescription sets. This is done by first finding redescription set with the ReReMi, and then forcing the same size of the redescription set on the CLUS-RM, since it produces much more redescriptions than the ReReMi algorithm.

We divided the results based on the operators allowed for query construction. In the first experiment we allow using disjunctions, conjunctions, negations and in the second experiment only conjunctions. For the ReReMi, we used *max product buckets=200*, *max number of pairs = 500* when using all logical operators, *max number of pairs = 1000* when using only conjunctions. Also, we allowed a maximum of 15 variables for each query. Redescriptions were required to have the maximal $p - value$ of 0.01, the minimal Jaccard coefficient of 0.5 and the minimal support of 5 elements. After obtaining redescriptions with the ReReMi algorithm, we used the *Filter redundant redescriptions* option to remove duplicate and redundant redescriptions with the *max overlap option* equal to 0.99. For each redescription set, we optimized a redescription set of the same size by using the CLUS-RM algorithm. We used 800 iterations keeping constraints for the Jaccard coefficient, the $p-value$ and support. Maximum allowed average tree depth was set to 8 and we used the complete randomization in the initialization procedure.

For the generated redescription sets, we plot comparative boxplots for the Jaccard coefficient, the log_{10} of the $p-value$, the element overlap, the attribute overlap and the rule size. The element overlap is the average Jaccard coefficient of covered elements by one redescription with respect to all other redescriptions in the redescription set, similarly the attribute overlap is the average Jaccard coefficient of the attributes contained in the redescription queries compared to every other redescription in the set. To emphasize importance of the redescription size from the point of understandability ($|attr(R)| \geq 20$ considered to be highly complex to understand), we calculate the normalized redescription size as follows:

$$R_{size} = \begin{cases} \frac{|attr(R)|}{20} & , |attr(R)| < 20 \\ 1 & , 20 \leq |attr(R)| \end{cases}$$

To obtain comparative results, we optimized JS_{pes} with ReReMi algorithm and then recalculated the score for each redescription to obtain JS_m.

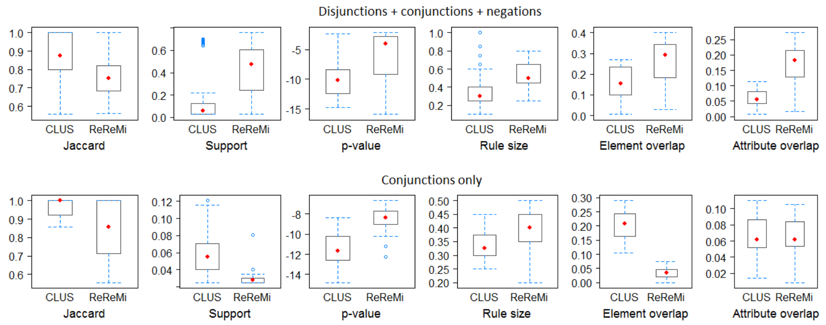


Fig. 4: The CLUS-RM and the ReReMi algorithm comparison on two redescription sets: constructed by using disjunctions, conjunctions, negations (120 redescriptions) and by using only conjunction operator (36 redescriptions)

The Figure 4 shows that the CLUS-RM found statistically significant redescriptions with Jaccard coefficient higher than those produced by the ReReMi algorithm. Due to its goal of finding highly accurate but minimally overlapping redescriptions in terms of elements and attributes, it found redescriptions with smaller support when conjunctions, disjunctions and negations are allowed. One important thing is that this was achieved by using redescriptions that mostly have smaller query size than the ReReMi produced redescriptions. We report two more statistics, the element coverage (the percentage of total elements described by at least one redescription) and attribute coverage (the percentage of attributes used in redescription rules). The CLUS-RM described 99% of elements while ReReMi described 100% elements. The CLUS-RM used 47% of all attributes in the rules and the ReReMi used 41%.

The evaluation on the redescription sets constructed by only using conjunction operator showed that the CLUS-RM produced redescriptions with higher Jaccard coefficient, higher support and smaller $p - value$ than the ReReMi algorithm. As a consequence, the CLUS-RM has higher element overlap but also somewhat smaller query size in redescriptions. Attribute overlaps are comparable between approaches. The CLUS-RM covered 25% elements and the ReReMi algorithm 53% while the attribute coverage is 27% and 36%.

The CLUS-RM approach produces redescriptions containing mainly conjunction operators while the ReReMi approach uses mostly disjunctions if allowed. The redescription sets obtained with the CLUS-RM contained highly accurate, statistically significant, mostly non overlapping redescriptions. There are two possible techniques available to obtain redescriptions with higher support with the CLUS-RM algorithm: to increase the minimal support or to increase the redescription set size. We believe that the proposed approach complements the ReReMi approach by finding many significant conjunction based redescriptions.

6 Conclusion

This work introduces a novel redescription mining framework which optimizes a redescription set of user defined size. The algorithm is based on multi-target predictive clustering trees, which allows using element coverage by rules constructed on one view as targets for the other view. Produced redescriptions incrementally improve the redescription set by using a predefined set of criteria (the Jaccard coefficient, the p-value, the element overlap and the attribute overlap). The ability to construct many different redescriptions and use them to optimize a set of fixed size differentiates the approach from currently proposed solutions. We analysed the algorithm time complexity and measured its performance on data describing world countries. The results show that, when finding redescriptions containing only conjunction operator, there are benefits of using more iterations. Generated redescriptions are statistically relevant with p-values less than 10^{-5}. Many generated rules contained the maximum of 6 attributes per rule in a redescription. Finally, we compare some characteristics of redescription sets generated by the CLUS-RM and the ReReMi algorithms. These results and comparison reveal the main difference in algorithm preference - CLUS-RM producing more accurate redescriptions using much more conjunctive rules.

In future work, we plan to extend the current framework by deploying Random Forest of PCTs, which should further boost resulting redescription sets in terms of size, diversity and quality. We also intend to work on more comprehensive and objective evaluation of redescription sets.

Acknowledgement

The authors would like to acknowledge the European Commission's support through the MAESTRA project (Gr. no. 612944), the MULTIPLEX project (Gr.no. 317532), the InnoMol project (Gr. no. 316289), and support of the

Croatian Science Foundation (Pr. no. 9623: Machine Learning Algorithms for Insightful Analysis of Complex Data Structures).

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining Association Rules Between Sets of Items in Large Databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207-216, Washington, D.C. (1993)
2. Bickel, S., Scheffer, T.: Multi-View Clustering. In Proceedings of the Fourth IEEE International Conference on Data Mining, pp. 19-26, Washington. (2004)
3. Blockeel., H.: Top-down Induction of First Order Logical Decision Trees. Phd thesis, Katholieke Universiteit Leuven, Department of Computer Science. (1998)
4. Cohen, E., Datar, M., Fujiwara, S., Gionis, A., Indyk, P., Motwani, R., Ullman, J., D., Yang, C.: Finding interesting associations without support pruning. In ICDE, pp. 489-499. (2000)
5. Galbrun, E., Kimmig, A.: Finding relational redescriptions. Machine Learning, pp. 225-248. (2014)
6. Galbrun, E., Miettinen, P.: From black and white to full color: extending redescription mining outside the Boolean world. Statistical Analysis and Data Mining, pp. 284-303. (2012)
7. Galbrun, E. and Miettinen, P. Siren: An Interactive Tool for Mining and Visualizing Geospatial Redescriptions. KDD, pp. 1544-1547. (2012)
8. Galbrun, E., Miettinen, P.: A Case of Visual and Interactive Data Analysis: Geospatial Redescription Mining. Instant Interactive Data Mining Workshop @ ECML-PKDD. (2012)
9. Galbrun, E.: Methods for Redescription mining. Phd thesis, University of Helsinki. (2013)
10. Gallo, A., Miettinen, P., Mannila, H.: Finding Subgroups having Several Descriptions: Algorithms for Redescription Mining. In Proceedings of the SIAM International Conference on Data Mining, pp. 334-345, Atlanta, Georgia. (2008)
11. Gamberger, D., Mihelčić, M., Lavrač, N., Multilayer Clustering: A Discovery Experiment on Country Level Trading Data. In Proceedings of the 17th International Conference on Discovery Science, Lecture Notes in Computer Science, pp. 87-98, Bled. (2014)
12. Gamberger, D., Lavrač N., Expert-Guided Subgroup Discovery: Methodology and Application. Journal of Artificial Intelligence Research, 17, pp. 501-527. (2002)
13. Giacometti, A., Li, D. H., Marcel, P., Soulet, A.: 20 Years of Pattern Mining: A Bibliometric Survey. SIGKDD Explor. Newsl., pp. 41-50. (2014)
14. Kocev, D., K., Vens, C., Struyf, J., Džeroski, S.: Tree ensembles for predicting structured outputs. Pattern Recognition, pp. 817-833. (2013)
15. Lavrač, N., Kavšek, B., Flach, P., and Todorovski Lj.: Subgroup Discovery with CN2-SD, J. Mach. Learn. Res., 5, pp. 153-188. (2004)
16. Mihelčić, M., Džeroski S., Lavrač N., Šmuc. T.: Redescription mining with multilabel Predictive Clustering Trees. In Proceedings of the fourth workshop on New Frontiers in Mining Complex Patterns @ ECML-PKDD, pp. 86-97, Porto. (2015)
17. Parida, L., Ramakrishnan, N.: Redescription Mining: Structure Theory and Algorithms. In Proceedings of the 20th National Conference on Artificial Intelligence, pp. 837-844, Pittsburgh, Pennsylvania (2004)
18. Piccart, B.: Algorithms for Multi-Target Learning. Phd thesis, Katholieke Universiteit Leuven. (2012)

19. Ramakrishnan, N., Kumar, D., Mishra, B., Potts, M., Helm, R. F.: Turning CARTwheels: an alternating algorithm for mining redescriptions. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 266-275, Seattle, WA. (2004)
20. Stojanova, D., Ceci, M., Appice, A., Džeroski, S.: Network regression with predictive clustering trees. Data Mining and Knowledge Discovery, pp. 378-413. (2012)
21. UNCTAD database, `http://unctadstat.unctad.org/EN/`.
22. World Bank database, `http://data.worldbank.org/`.
23. Zaki, M. J., and Ramakrishnan, N. Reasoning about sets using redescription mining. In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 364-373, Chicago, Illinois. (2005)
24. Zinchenko, T., Redescription Mining Over non-Binary Data Sets Using Decision Trees. Masters thesis, Universität des Saarlandes. (2014)

Appendix

We present several shorter redescriptions mined by the CLUS-RM and the ReReMi algorithm. The full names of the attributes used in redescription queries can be seen in Figure 5.

In Table A.1 we show two very accurate redescriptions mined with the ReReMi algorithm and compare it to two redescriptions mined with the CLUS-RM.

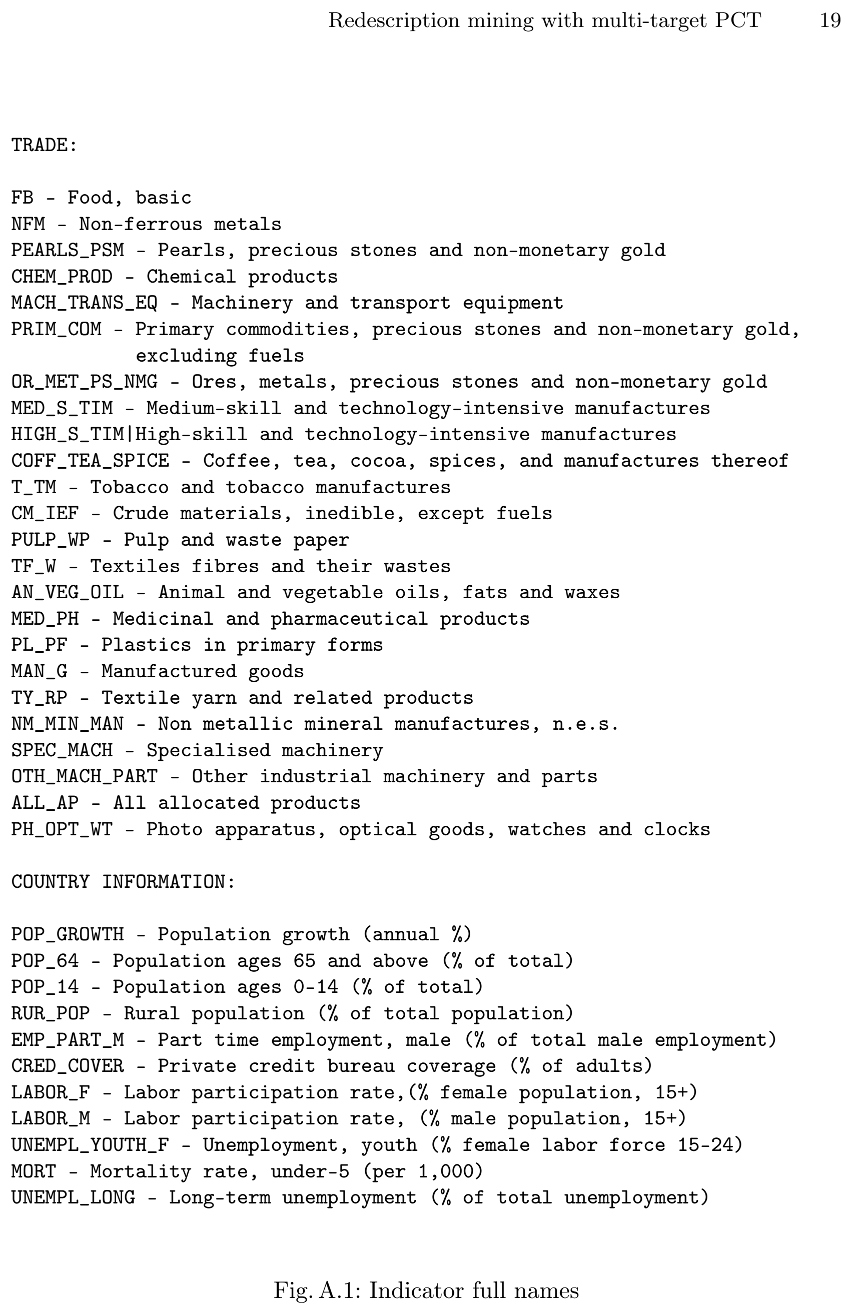
In Table A.2, we present two redescriptions containing conjunction and disjunction operator obtained with the ReReMi algorithm, and two redescriptions containing conjunctions and negations obtained with the CLUS-RM algorithm. This examples demonstrate the main difference between the methodologies. The ReReMi algorithm uses disjunction operator often in redescription construction whereas the CLUS-RM mostly uses conjunction operator to construct redescriptions.

Table A.1: Redescription examples produced by CLUS-RM and ReReMi algorithm using only conjunction operator

Redescriptions	JS	supp	p-value	Algorithm
$-0.04 \leq POP\_GROWTH \leq 2.49 \land$ $14.05 \leq POP\_64 \leq 21.1 \land$ $13.74 \leq RUR\_POP \leq 50.1 \land$ $4.7 \leq EMP\_PART\_M \leq 11.7$ $1.0 \leq E\_PL\_PF \leq 2.0 \land$ $0.65 \leq E/I\_SPEC\_MACH \leq 4.31 \land$ $18.0 \leq I\_MED\_S\_TIM \leq 25.0$	1.0	14	$1.5 \cdot 10^{-13}$	CLUS-RM
$26.0 < RUR\_POP \land$ $31.0 < CRED\_COVER < 47.5 \land$ $52.9 < LABOR\_F < 67.5$ $4.0 < E\_MAN\_G < 17.0 \land$ $17.0 < I\_MED\_S\_TIM < 33.0 \land$ $I\_TF\_W < 0.0 \land I\_AN\_VEG\_OIL < 1.0 \land$ $I\_TY\_RP < 2.0 \land$ $0.01 < E/I\_MED\_PH < 0.24$	1.0	8	$6.4 \cdot 10^{-11}$	ReReMi
$14.56 \leq POP\_14 \leq 21.54 \land$ $2.2 \leq MORT \leq 4.5 \land$ $61.8 \leq LABOR\_M \leq 68.1$ $1.0 \leq E/I\_OTH\_MACH\_PART \leq 1.62 \land$ $14.0 \leq I\_HIGH\_S\_TIM \leq 28.0 \land$ $0.74 \leq E/I\_PULP\_WP \leq 13.04$	1.0	9	$1.9 \cdot 10^{-11}$	CLUS-RM
$14.57 < POP\_14 < 14.98 \land$ $POP\_GROWTH < 0.26 \land$ $24.0 < UNEMPL\_YOUTH\_F < 44.3$ $9.0 < E\_FB < 14.0 \land$ $1.0 < E\_NFM \ \land \ I\_T\_TM < 0.0 \land$ $2.17 < E/I\_PEARLS\_PSM < 885.93 \land$ $0.83 < E/I\_PRIM\_COM < 2.93$	1.0	5	$4.6 \cdot 10^{-9}$	ReReMi

Table A.2: Redescription examples produced by CLUS-RM and ReReMi algorithm using conjunction, disjunction and negation operators

Redescriptions	JS	supp	p-value	Algorithm
$8.0 \leq MORT \leq 181.6$ $\neg\ (0.65 \leq E/I\_SPEC\_MACH \leq 61.94)$	0.842	139	$2.6 \cdot 10^{-4}$	CLUS-RM
$4.9 < MORT \ \vee\ 22.8 < POP\_14 \vee$ $-0.26 < POP\_GROWTH < -0.09 \vee$ $11.1 < UNEMPL\_LONG$ $E\_PH\_OPT\_WT < 0.0 \wedge$ $E/I\_SPEC\_MACH < 0.65$	0.865	148	$6.3 \cdot 10^{-4}$	ReReMi
$-1.48 \leq POP\_GROWTH \leq 0.48 \wedge$ $2.9 \leq MORT \leq 5.4 \wedge$ $-1.49 \leq BAL \leq 10.11 \wedge$ $4.24 \leq STOCKS \leq 166.61$ $1.0 \leq I\_NM\_MIN\_MAN \leq 1.0 \wedge$ $0.81 \leq E/I\_OTH\_MACH\_PART \leq 3.15 \wedge$ $19.0 \leq I\_MACH\_TRANS\_EQ \leq 32.0 \wedge$ $0.92 \leq E/I\_CHEM\_PROD \leq 6.66$	1.0	12	$8.56 \cdot 10^{-13}$	CLUS-RM
$MORT < 95.5 \wedge$ $75.9 < LABOR\_F \wedge$ $1.89 < POP\_GROWTH$ $((94.0 < I\_ALL\_AP < 99.0 \wedge$ $E/I\_PL\_PF < 0.03) \vee$ $31.0 < I\_OR\_MET\_PS\_NMG < 34.0 \vee$ $5.38 < E/I\_CM\_IEF < 7.81) \wedge$ $1.0 < E/I\_COFF\_TEA\_SPICE$	1.0	10	$6.3 \cdot 10^{-12}$	ReReMi



```
TRADE:

FB - Food, basic
NFM - Non-ferrous metals
PEARLS_PSM - Pearls, precious stones and non-monetary gold
CHEM_PROD - Chemical products
MACH_TRANS_EQ - Machinery and transport equipment
PRIM_COM - Primary commodities, precious stones and non-monetary gold,
           excluding fuels
OR_MET_PS_NMG - Ores, metals, precious stones and non-monetary gold
MED_S_TIM - Medium-skill and technology-intensive manufactures
HIGH_S_TIM|High-skill and technology-intensive manufactures
COFF_TEA_SPICE - Coffee, tea, cocoa, spices, and manufactures thereof
T_TM - Tobacco and tobacco manufactures
CM_IEF - Crude materials, inedible, except fuels
PULP_WP - Pulp and waste paper
TF_W - Textiles fibres and their wastes
AN_VEG_OIL - Animal and vegetable oils, fats and waxes
MED_PH - Medicinal and pharmaceutical products
PL_PF - Plastics in primary forms
MAN_G - Manufactured goods
TY_RP - Textile yarn and related products
NM_MIN_MAN - Non metallic mineral manufactures, n.e.s.
SPEC_MACH - Specialised machinery
OTH_MACH_PART - Other industrial machinery and parts
ALL_AP - All allocated products
PH_OPT_WT - Photo apparatus, optical goods, watches and clocks

COUNTRY INFORMATION:

POP_GROWTH - Population growth (annual %)
POP_64 - Population ages 65 and above (% of total)
POP_14 - Population ages 0-14 (% of total)
RUR_POP - Rural population (% of total population)
EMP_PART_M - Part time employment, male (% of total male employment)
CRED_COVER - Private credit bureau coverage (% of adults)
LABOR_F - Labor participation rate,(% female population, 15+)
LABOR_M - Labor participation rate, (% male population, 15+)
UNEMPL_YOUTH_F - Unemployment, youth (% female labor force 15-24)
MORT - Mortality rate, under-5 (per 1,000)
UNEMPL_LONG - Long-term unemployment (% of total unemployment)
```

Fig. A.1: Indicator full names

Details of redescription mining augmented with Random Forest of Predictive Clustering Trees are described in the following publication (included in this chapter):

M. Mihelčić, S. Džeroski, N. Lavrač, and T. Šmuc, "Redescription mining augmented with random forest of multi-target predictive clustering trees," *Journal of Intelligent Information Systems*, pp. 1–34, 2017, In press.

This work is an extended version of *Redescription Mining with Multi-target Predictive Clustering Trees* paper. The additional authors contributions are as follows. Matej Mihelčić devised and implemented the extensions required to incorporate random forest of Predictive Clustering trees in the CLUS-RM algorithm, devised and implemented extensions required to alleviate constraints on redescription set size in redescription set optimization procedure, computed the computational time complexity of the extended algorithm, performed and analysed all experiments, incorporated the significance and distinct coverage score in the optimization procedure, wrote the majority of text related to the newly introduced extensions and actively participated in writing revisions. Nada Lavrač suggested performing research in the field of redescription mining. Tomislav Šmuc pointed out the benefits of using coverage as one of the redescription set optimization measures and assisted in defining the optimization measure based on redescription p-value. Tomislav Šmuc, Nada Lavrač and Sašo Džeroski participated in writing, proof-reading and correcting the text of the manuscript and the revisions.

Noname manuscript No.
(will be inserted by the editor)

Redescription mining augmented with Random Forest of multi-target Predictive Clustering Trees

Matej Mihelčić · Sašo Džeroski · Nada Lavrač · Tomislav Šmuc

Received: date / Accepted: date

Abstract In this work, we present a redescription mining algorithm that uses Random Forest of Predictive Clustering Trees (RFPCTs) for generating and iteratively improving a set of redescriptions. The approach uses information about element membership in different queries, generated from a single constructed PCT, to explore redescription space, while queries obtained from the Random Forest of PCTs increase candidate diversity. The approach is able to produce highly accurate, statistically significant redescriptions described by Boolean, nominal or numerical attributes. As opposed to current tree-based approaches that use multiclass or binary classification, we explore the benefits of using multi label classification and multi target regression to create redescriptions. Major benefit of the approach, compared to other state of the art solutions, is that it does not require specifying minimal threshold on redescription accuracy to obtain highly accurate, optimized set of redescriptions. The process of Random Forest based augmentation and different modes of redescription set creation are evaluated on three datasets

Matej Mihelčić and Tomislav Šmuc
Ruđer Bošković Institute
Bijenička cesta 54, 10000 Zagreb, Croatia
{matej.mihelcic, tomislav.smuc}@irb.hr

Sašo Džeroski and Nada Lavrač
Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
{saso.dzeroski, nada.lavrac}@ijs.si

Matej Mihelčić, Sašo Džeroski, Nada Lavrač
Jožef Stefan International Postgraduate School
Jamova cesta 39, 1000 Ljubljana, Slovenia

This is a post-peer-review, pre-copy edit version of an article published in Journal of Intelligent Information Systems. The final authenticated version is available online at: https://doi.org/10.1007/s10844-017-0448-5.

with different properties. We use the same datasets to compare the performance of our algorithm to state of the art redescription mining approaches.

Keywords knowledge discovery, redescription mining, random forest, predictive clustering trees, world countries, computer science bibliography, bioclimatic niches

1 Introduction

Pattern mining [1, 15, 16, 24] aims at discovering descriptive rules learned from data. Redescription mining (RM) [27] shares this goal and is directed towards finding different descriptions of patterns by using one or more disjoint sets of descriptive attributes (these disjoint sets are also called views). The input to redescription mining algorithms consists of one or more tables containing all attributes from the given view and their corresponding values for all elements contained in the dataset. One input example obtained from the DBLP database [6, 11], containing information about authors of scientific papers, can be seen in Table 1.

Table 1: Input example for the DBLP dataset.

(a) View 1: Author-conference bipartite graph

Entity	*ISAAC*	*FCT*	...	*PLILP*
J.D.Tygar	false	false	...	false
.	.	.	...	.
.	.	.	...	.
.	.	.	...	.
JohnH.Reif	true	true	...	false
.	.	.	...	.
.	.	.	...	.
.	.	.	...	.
ChrisClifton	true	false	...	false
.	.	.	...	.
.	.	.	...	.
.	.	.	...	.
AthenaVakali	false	false	...	false

(b) View 2: Co-authorship network (self-authorship excluded)

Entity	*J.D.Tygar*	*AdolfyHoisie*	...	*AthenaVakali*
J.D.Tygar	false	false	...	false
.	.	.	...	.
.	.	.	...	.
.	.	.	...	.
JohnH.Reif	true	false	...	false
.	.	.	...	.
.	.	.	...	.
.	.	.	...	.
ChrisClifton	true	false	...	false
.	.	.	...	.
.	.	.	...	.
.	.	.	...	.
AthenaVakali	false	false	...	false

Redescription mining is an unsupervised, descriptive knowledge discovery task. Similarly as Association rule mining [1], it discovers associations between attributes. However, instead of finding one-directional implication relations it finds bi-directional equivalence relations. Association discovery in two-view data [31] finds both uni and bi-directional associations but is aimed at explaining how these two views are related. Redescription mining is related to multi-view [2] and multi-layer [13] clustering, though the main goal here is to find accurate redescriptions of interesting subsets of data, while clustering tends to find clusters that are not always easy to interpret. Finding similarities between different elements and connections between different descriptive attribute sets (views) ultimately leads to better understanding of the underlying data. The output of redescription mining is a set of redescriptions which are tuples of rules (logical formulas). The aim is to make these rules understandable and interpretable.

We present one redescription example $R_{ex} = (q_{1_{ex}}, q_{2_{ex}})$, where $q_{1_{ex}} = \neg$UAI $\wedge$ $\neg$PKDD $\wedge$ SEBD $\wedge$ SIGMOD $\wedge$ LPNMR and $q_{2_{ex}}$ = Thomas Eiter $\wedge$ Gianluigi Greco. For all authors described by the redescription R_{ex} it must be valid that they co-authored a paper with Thomas Eiter and co-authored a (not necessarily the same) paper with Gianluigi Greco. They have also published at least one paper on conferences SEBD, SIGMOD and LPNMR but have not published any papers in conferences PKDD and UAI. Interestingly, Thomas Eiter and Gianluigi Greco have co-authored the paper *Boosting Information Integration: The INFOMIX System* published in SEBD in 2005. This makes the group of described authors very well connected in terms of co-authorship.

Applications of redescription mining

Redescription mining is highly applicable in biology, economy, pharmacy, ecology and many other fields, where it is important to understand connections between different descriptors and to find regularities that are valid for different element subsets. For instance, it might be interesting to associate gene functions with gene locations in different genomes, to study similarities or differences in structure of different organisms, to relate proteins and different chemical compounds to understand the effects of interactions, with potential application in design of new and more effective medicines, to associate animal species habitats with weather locations, in order to obtain knowledge about the effects of these conditions on habitats and co-habitats of different animal species, or to associate authors of scientific papers with different scientific conferences to obtain groups of related authors sharing some area of research. In these applications, implication relations, as provided with association rule mining, are not strong enough to allow explaining the underlying phenomena. Due to strong equivalence relations it produces, redescription mining is well suited for relating a set of attributes, which are generally understood well (such as questionnaires and various written or motorical tests devised by researchers) to a set of attributes, containing different measurements (medical, biological) which are not always understood well.

Related work in redescription mining

The field of redescription mining was introduced in the work from Ramakrishnan et al. [27] which presents a novel decision tree - based redescription mining algorithm called the CARTwheels. The algorithm builds two decision trees (one for each view) that are joined in the leaves. Redescriptions are found by examining the paths from the root node of the first tree to the root node of the second and the algorithm uses multi class classification to guide the search between the two views. Other approaches for redescription mining include: the approach proposed by Zaki and Ramakrishnan [33] which uses a lattice of closed descriptor sets to find redescriptions, the approach proposed by Parida and Ramakrishnan [25] for mining exact and approximate redescriptions based on relaxation lattice. Further, Gallo et al. [12] present the greedy algorithm and the MID (Mining Interesting Descriptors) algorithm based on frequent itemset mining.

Galbrun and Miettinen [8] extend the greedy approach by Gallo et al. [12] to work on numerical data, thus increasing capabilities of redescription mining algorithms. Galbrun and Kimming extend redescription mining to a relational [7] setting, while Galbrun and Miettinen make extensions to allow interactive redescription mining [10]. Recently, two novel tree-based algorithms were proposed by Zinchenko [34]. These approaches use decision trees in a non-Boolean setting and present different methods of layer by layer tree construction, which allows making informed splits based on nodes at each level of the tree.

Methodology

In this work, we present a multi-target predictive clustering trees (PCTs) [3, 19] based redescription mining algorithm developed to create a large number of diverse redescriptions. As in our previous work ([22]), all created PCTs use multi-target classification or regression to find highly accurate, statistically significant redescriptions, which differentiates it from other tree based approaches, especially the CARTwheels approach. Using multi-target PCTs allows us to build one model to find multiple redescriptions using nodes at all levels of the tree and due to inductive transfer [26], multi-target trees can outperform single label classification or regression trees. Each node in one separately created PCT model, used to guide the search, represents a separate rule that is used in the construction of a PCT from the opposite view. Generated redescriptions are used to iteratively improve and expand a redescription set of user suggested, not necessarily fixed size (which alleviates the hard constraints on redescription set size used in [22]). The algorithm presented in [22] has been extended to incorporate the random forest of PCTs as an augmenting model. This increases the accuracy and diversity of produced redescriptions. The approach relies on the fact that a great number of PCTs can be trained and converted to rules in parallel. Thus, this augmented process can be executed very efficiently and almost at the same running time, if executed in parallel threads, as the process containing only one PCT. Additional benefit of the approach is that it allows creation of highly optimized redescription sets without requiring users to constrain redescription accuracy. This is an advantage, compared to current state of the art approaches, because it usually needs to be determined through experimentation. Finally, rule minimization procedure, presented in our previous work ([22]), allows reducing the number of attributes that describe a given pattern without changing redescription accuracy or support. This allows obtaining shorter rules even when using trees of bigger depth size.

Structure

After introducing the necessary notation (Section 2), we present the extended algorithm and perform the run-time complexity analysis of redescription mining process (Section 3). In Section 4, we evaluate algorithm extensions, compare its performance with several state of the art approaches on three datasets with different properties and present redescription examples obtained by these approaches. Finally, we conclude and outline directions for future work in Section 5.

2 Notation and definitions

Redescription mining in general considers redescriptions constructed on a set of views $\{W_1, W_2, \dots, W_n\},\ n \geq 1$, however we use only two views $\{W_1, W_2\}$ since all current redescription mining approaches use maximally two views. Using more than two views significantly increases computational complexity and requires data containing several disjoint sets of attributes describing the same set of elements. The corresponding attribute (variable) sets are denoted by V_1 and V_2. Each view contains the same set of $|E|$ elements and two different sets of attributes of size $|V_1|$ and $|V_2|$. Value $W_1(i, j)$ is the value of element e_i for the attribute a_j in view W_1. The data $D = (V_1, V_2, E, W_1, W_2)$ is a quintuple of the attribute sets, the element set, and the appropriate view mappings. A query (denoted q) is a logical formula F containing attributes from V_1 or V_2 as variables and the set of elements described by a query is called its support. A redescription $R = (q_1, q_2)$ is defined as a pair of queries, one for each view in the data and its support is the set of elements supported by both queries that constitute this redescription: $\mathit{supp}(R) = \mathit{supp}(q_1) \cap \mathit{supp}(q_2)$. We use $\mathit{attr}(R)$ to denote the multiset of attributes used in the redescription R and $\mathit{attrs}(R)$ to denote the corresponding set of attributes. The accuracy of a redescription $R = (q_1, q_2)$ is measured with the Jaccard index (Jaccard similarity coefficient):

$$J(R) = \frac{|\mathit{supp}(q_1) \cap \mathit{supp}(q_2))|}{|\mathit{supp}(q_1) \cup \mathit{supp}(q_2)|}$$

The Jaccard index is not the only measure used in the field because it is possible to obtain redescriptions with large support for which it is highly probable to have very good overlap of their queries. In this cases it is preferred to have redescriptions that reveal some more specific knowledge about the studied problem that is harder to obtain by random sampling from the underlying data distribution. This is why we compute the statistical significance (p-value) of each obtained redescription. The marginal probability of a query q_1, q_2 is denoted as $p_1 = \frac{|\mathit{supp}(q_1)|}{|E|}$ and $p_2 = \frac{|\mathit{supp}(q_2)|}{|E|}$ respectively. We define the set of elements in the intersection of the queries with $o = \mathit{supp}(q_1) \cap \mathit{supp}(q_2)$. The corresponding p-value [11] is defined as

$$pV(q_1, q_2) = \sum_{n=|o|}^{|E|} \binom{|E|}{n} (p_1 \cdot p_2)^n \cdot (1 - p_1 \cdot p_2)^{|E|-n}$$

The p-value tells us if we can dismiss the null hypothesis that assumes that we obtained a given subset of elements by joining two random rules with marginal probabilities equal to the fraction of covered elements. If the obtained p-value is lower than some predefined threshold, called the significance level, then this null hypothesis should be rejected. This estimate is optimistic when the assumption that all elements can be sampled with equal probability does not hold (which is often the case in practice).

We use two redescription quality measures based on properties of a redescription set that contains them. These measures, created with similar intuitions to those presented by Knobe and Ho ([18]), provide information about the level of redundancy of a given redescription with respect to described elements and attributes used in redescription queries in every other redescription contained in the given redescription set. The measure providing information about the redundancy of elements contained in the redescription support is called the average redescription element Jaccard index and is defined as:

$$AEJ(R_i) = \frac{1}{|\mathcal{R}|-1} \cdot \sum_{j=1}^{|\mathcal{R}|} J(supp(R_i), supp(R_j)),\ i \neq j$$

Analogously, the measure providing information about the redundancy of attributes contained in redescription queries, called the average redescription attribute Jaccard index, is defined as:

$$AAJ(R_i) = \frac{1}{|\mathcal{R}|-1} \cdot \sum_{j=1}^{|\mathcal{R}|} J(attrs(R_i), attrs(R_j)),\ i \neq j$$

To emphasize importance of the redescription size from the point of understandability ($|attr(R)| \geq 20$ considered to be highly complex to understand), we calculate the normalized redescription size as follows:

$$R_{size} = \begin{cases} \frac{|attr(R)|}{20} & , |attr(R)| < 20 \\ 1 & , 20 \leq |attr(R)| \end{cases}$$

3 The CLUS-RM algorithm

In this section, we describe the algorithm for mining redescriptions named CLUS-RM. The algorithm optimizes a redescription set of a size determined by redescription properties or suggested by a user. It uses multi-target predictive clustering trees (PCTs) [3,19] to create a cluster hierarchy that is used to explore the redescription space. In addition, a random forest of predictive clustering trees is used to diversify the search and to increase the overall redescription accuracy. We start by explaining the pseudo code of the algorithm (Algorithm 1) and then go into details of each procedure in the algorithm.

The algorithm starts by creating initial clusters for both views (line 2 and 3 in Algorithm 1) which is achieved by transforming a non-labeled dataset into a labeled dataset of positive, original elements (elements originally present in the dataset) and artificially generated, negative elements (elements not originally present in the dataset but artificially constructed and added to the dataset). For each element in the original view, we construct one negative, synthetic element (see Figure 2) in such a way so that the original correlations among the attributes are broken. We achieve this by random shuffling of attribute values between the elements. The procedure allows experimentation with the number of shuffling steps and the number of attributes that are copied from the original elements to the artificial element. Complete randomization is achieved when the number of shuffling steps equals the number of attributes in the dataset and exactly one attribute value is copied to the artificial element at each step from a randomly chosen original element. The original elements are assigned a target label of 1.0, while the artificial elements are assigned a target label of 0.0 (see Table 2). Target label is used to give information to a supervised learning algorithm, such as PCT, which elements were originally present in the data and which were artificially constructed. The

Algorithm 1 The CLUS-RM algorithm

```
Input: First view data ($W_1$), Second view data ($W_2$), Settings file
Output: A set of redescriptions $\mathcal{R}$
 1: procedure CLUS-RM
 2:     [$\mathcal{D}_{W_1init}$, $\mathcal{D}_{W_2init}$] ← prepareTargetsForInitialPCT($W_1$,$W_2$)
 3:     [PCT$W_1$, PCT$W_2$] ← createSidesInitialPCT($\mathcal{D}_{W_1init}$, $\mathcal{D}_{W_2init}$)
 4:     [R$W_1$, R$W_2$] ← extractRules(PCT$W_1$, PCT$W_2$)
 5:     initializeArrays(elFreq, attrFreq, redScoreEl, redScoreAt, numEx, numAttr,
                              numRetRed)
 6:     while RunInd<maxIter do
 7:         [TmpR$W_1$, TmpR$W_2$] ← emptyRuleSet()
 8:         [SR$W_1$, SR$W_2$] ← emptyRuleSet()
 9:         [$\mathcal{D}_{W_1Targ}$, $\mathcal{D}_{W_2Targ}$] ← prepareTargets(R$W_2$, R$W_1$)
10:         [PCT$W_1$, PCT$W_2$] ← createPCT($\mathcal{D}_{W_1Targ}$, $\mathcal{D}_{W_2Targ}$)
11:         [RFPCT$W_1$, RFPCT$W_2$] ← createSRFPCT($\mathcal{D}_{W_1Targ}$, $\mathcal{D}_{W_2Targ}$)
12:         TmpR$W_1$ ← TmpR$W_1$ $\cup_*$ extractRules(PCT$W_1$)
13:         TmpR$W_2$ ← TmpR$W_2$ $\cup_*$ extractRules(PCT$W_2$)
14:         SR$W_1$ ← SR$W_1$ $\cup_*$ extractRules(RFPCT$W_1$)
15:         SR$W_2$ ← SR$W_2$ $\cup_*$ extractRules(RFPCT$W_2$)
16:         R$W_1$ ← R$W_1$ $\cup_*$ TmpR$W_1$
17:         R$W_2$ ← R$W_2$ $\cup_*$ TmpR$W_2$
18:         $\mathcal{R}$ ← MineRed(R$W_1$, R$W_2$, SR$W_1$, SR$W_2$, expansionType,
                ConstSet, iteration, opSet, elFreq, attrFreq, redScoreEl, redScoreAt)
19:     $\mathcal{R}$ ← $minimizeReds(\mathcal{R})$
20:     return $\mathcal{R}$
```

division between the original and the artificial elements (the idea previously used in the work from Gamberger et al. [13]), allows us to construct a cluster hierarchy, simultaneously creating descriptions of the original elements. The described procedure is one possible way to construct the initial clusters; other approaches include assigning a random target attribute or using clusters computed by some other clustering algorithm. However, the initialization procedure used in our algorithm should preserve any strong (specific) connections and correlations that exist in the original data which are broken by using an approach that assigns random target labels.

After creating the initial dataset, we build predictive clustering trees on both views by performing regression on the target label and using other attributes as descriptive (line 3 in Algorithm 1). The decision to use regression trees instead of decision trees is purely technical, since it generates more rules because of the additional threshold associated with the target variable. These trees are converted to rules (line 4 in Algorithm 1) that describe element sets and are necessary for the next step of the algorithm. The rule lists RW_1 and RW_2 contain generated rules, and a new rule is added to the list if it differs from all other rules in a predefined number of attributes or if it describes a new unique element subset (the $\cup_*$ operator in Algorithm 1). The iterative process of the algorithm begins right after rule creation (line 6 in Algorithm 1). Here, we create targets based on the rules obtained in the previous step or in the initialization step (line 9 in Algorithm 1). The rules obtained by predictive clustering on W_1 are used to build targets for clustering on W_2 (denoted W_1T_1, W_1T_2), and vice versa. For each element in the dataset we assign label 1.0 if the element is described by some specific rule, otherwise 0.0 (see Table 3). For example, the attribute W_2T_1 from dataset for view 1 represents the condition $IF\ W_2A_1 = TRUE$ (constructed on dataset for view 2), which describes elements E_1, E_2, E_4, E_5. By placing this

Table 2: Creation of artificial elements for the random initialization procedure. For example, the artificial element E_1' in view 1 is created by copying a value for attribute W_1A_1 from original element E_4, for attribute W_1A_2 from E_1 and for attribute W_1A_3 from E_5. Since the element E_1' is artificially created, it is assigned a target value 0.0.

(a) Original dataset for view 1

Entity	W_1A_1	W_1A_2	W_1A_3
E_1	1.1	2.5	3.4
E_2	1.5	2.2	4.0
E_3	5.5	-0.6	-0.2
E_4	4.4	-0.2	2.0
E_5	3.2	1.7	2.9

(b) Original dataset for view 2

Entity	W_2A_1	W_2A_2	W_2A_3
E_1	TRUE	FALSE	FALSE
E_2	TRUE	TRUE	FALSE
E_3	FALSE	FALSE	TRUE
E_4	TRUE	TRUE	TRUE
E_5	TRUE	FALSE	TRUE

(c) Initial dataset for view 1

Entity	W_1A_1	W_1A_2	W_1A_3	Target
E_1	1.1	2.5	3.4	1.0
E_2	1.5	2.2	4.0	1.0
E_3	5.5	-0.6	-0.2	1.0
E_4	4.4	-0.2	2.0	1.0
E_5	3.2	1.7	2.9	1.0
E_1'	4.4	2.5	2.9	0.0
E_2'	3.2	-0.6	4.0	0.0
E_3'	3.2	-0.6	2.9	0.0
E_4'	4.4	-0.2	4.0	0.0
E_5'	5.5	1.7	2.9	0.0

(d) Initial dataset for view 2

Entity	W_2A_1	W_2A_2	W_2A_3	Target
E_1	TRUE	FALSE	FALSE	1.0
E_2	TRUE	TRUE	FALSE	1.0
E_3	FALSE	FALSE	TRUE	1.0
E_4	TRUE	TRUE	TRUE	1.0
E_5	TRUE	FALSE	TRUE	1.0
E_1'	TRUE	FALSE	TRUE	0.0
E_2'	FALSE	FALSE	TRUE	0.0
E_3'	TRUE	TRUE	TRUE	0.0
E_4'	FALSE	TRUE	FALSE	0.0
E_5'	FALSE	FALSE	TRUE	0.0

target attribute in the view 1 dataset, we guide the PCT construction (lines 9 and 10 in Algorithm 1) to create a cluster containing and describing the same set of elements with descriptive variables of view 1 (a choice that satisfies this condition is $IF\ W_1A_3 > 0$).

Table 3: Intermediate generation of labels based on discovered rules.

(a) Dataset for view 1

E	W_1A_1	W_1A_2	W_1A_3	W_2T_1	W_2T_2
E_1	1.1	2.5	3.4	1.0	0.0
E_2	1.5	2.2	4.0	1.0	0.0
E_3	5.5	-0.6	-0.2	0.0	0.0
E_4	4.4	-0.2	2.0	1.0	0.0
E_5	3.2	1.7	2.9	1.0	1.0

(b) Dataset for view 2

E	W_2A_1	W_2A_2	W_2A_3	W_1T_1	W_1T_2
E_1	TRUE	FALSE	FALSE	0.0	1.0
E_2	TRUE	TRUE	FALSE	0.0	1.0
E_3	FALSE	FALSE	TRUE	1.0	0.0
E_4	TRUE	TRUE	TRUE	1.0	0.0
E_5	TRUE	FALSE	TRUE	1.0	1.0

Random forest of PCTs is constructed (line 11 in Algorithm 1) by using the same targets as to construct the PCT used to guide the search (line 10 in Algorithm 1). PCTs in the forest represent a set of weak learners trained on subspaces of attributes with the purpose of diversifying produced redescriptions and increasing their accuracy. The use of random forest has several advantages: 1) due to restricted size of the attribute subspace used to make a split, it is able to avoid local optima, 2) it explores much larger number of attribute associations (depending on the number of trees used in the forest) which is very important for produced redescriptions. The number of PCTs to be used in a random forest and the size of a random subspace are user defined parameters. The random subspace size is usually set to $\sqrt{N}$ or $log_2(N)$ in predictive tasks, where N equals the number of attributes contained in the selected view. However, in redescription mining it is important to discover different attribute interactions. Thus it is useful to have guarantees on attribute membership in different random subspaces. We have computed the necessary size of a random subspace, given a random forest of PCT with defined parameters, so that an arbitrary attribute occurs with a given probability in at least one split of every tree in the forest. The subset size is computed as:

$k = N \cdot (1 - \sqrt[q]{1-p})$, where $q = (2^d - 1)$, d equals the average PCT depth and p denotes the desired probability to evaluate an arbitrary attribute in at least one split of every PCT in a random forest of given properties. Since, for very small number of attributes, this number quickly drops to 0, the subset size equals $k = max(\lceil N \cdot (1 - \sqrt[q]{1-p})\rceil, \lceil log_2(N)\rceil)$. Assigning probability to attribute occurrence in at least one split of every tree in a forest allows influencing accuracy and diversity of queries used to produce redescriptions. High occurrence probability should be used on sparse datasets, when higher accuracy is required, while using lower probability on dense datasets increases diversity and brings computational advantage since smaller subsets need to be evaluated. Random forest of PCTs can be trained in parallel with minor loss in computation time, compared to the algorithm presented in our previous work ([22]). Rules obtained in the previous step are combined into redescriptions (line 18 in Algorithm 1) if they satisfy a given set of constraints $ConstSet$. It consists of minimal Jaccard index ($minJ$), maximum allowed p-value ($maxPval$), minimum and maximum support ($minSupp$, $maxSupp$) which have to be satisfied for a redescription to be considered as a candidate for the redescription set. The default value of 0.01 is used for p-value and a minimal support of 2 elements if corresponding parameters are not specified. Specifying the Jaccard index constraint is optional. After performing redescription creation and redescription set optimization, all queries produced by the random forest models are discarded. Finally, queries of the resulting redescriptions are minimized in line 19 of Algorithm 1 by using the query minimization procedure presented in our previous work ([22]).

3.1 The procedure for creating redescriptions

The algorithm for creating redescriptions from rules (Algorithm 2) joins view 1 rules (or their negation, if allowed by the user) with rules (or its negation) from view 2 (see Figure 1 and line 2 in Algorithm 2). We distinguish three cases of creating redescriptions from rules (expansion types):

1. Unguided initial: $UInit \leftarrow ((SRW_1 \cup RW_1) \times^{opSet\setminus\{\vee\}}_{ConstSet} (SRW_2 \cup RW_2))$
2. Unguided: $U \leftarrow ((SRW_1 \cup RW_{1_{newRuleIt}}) \times^{opSet\setminus\{\vee\}}_{ConstSet} (SRW_2 \cup RW_{2_{newRuleIt}}))$
3. Guided: $G \leftarrow ((SRW_1 \cup RW_{1_{newRuleIt}}) \times^{opSet\setminus\{\vee\}}_{ConstSet} RW_{2_{oldRuleIt}}) \cup$ $(RW_{1_{oldRuleIt}} \times^{opSet\setminus\{\vee\}}_{ConstSet} (SRW_2 \cup RW_{2_{newRuleIt}}))$

The $\times^{opSet}_{ConstSet}$ operator denotes a Cartesian product of two sets, allowing the use of logical operators from $opSet$ and leaving only those redescriptions that satisfy a given set of constraints $ConstSet$. The unguided expansion allows obtaining redescriptions with more diverse subsets of elements that can later be improved through the iteration process.

The algorithm finds first $numRed$ redescriptions if the size is fixed by the user, or max(20,$numRed$) redescriptions if the size is suggested or fully automatically determined (line 4 in Algorithm 2). This minimal number of redescriptions is used to provide a set which is not very large but still provides different information about the elements and contains enough redescriptions to perform statistical analysis. After the minimal number of distinct redescriptions is found, the set is iteratively improved by exchanging the redescription with the worst comparative score with

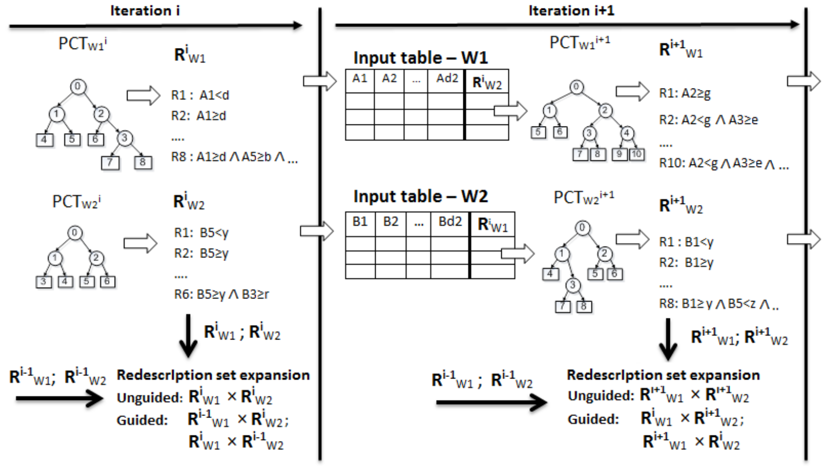

Iteration i
$PCT_{W1}{}^{i}$
R^{i}_{W1}
R1 : A1<d
R2: A1≥d
....
R8 : A1≥d ∧ A5≥b ∧ ...
$PCT_{W2}{}^{i}$
R^{i}_{W2}
R1: B5<y
R2: B5≥y
....
R6: B5≥y ∧ B3≥r
R^{i}_{W1} ; R^{i}_{W2}
R^{i-1}_{W1}; R^{i-1}_{W2}
Redescription set expansion
Unguided: $R^{i}_{W1} \times R^{i}_{W2}$
Guided: $R^{i-1}_{W1} \times R^{i}_{W2}$;
$R^{i}_{W1} \times R^{i-1}_{W2}$
Iteration i+1
Input table – W1
A1
A2
...
Ad2
R^{i}_{W2}
$PCT_{W1}{}^{i+1}$
R^{i+1}_{W1}
R1: A2≥g
R2: A2<g ∧ A3≥e
....
R10: A2<g ∧ A3≥e ∧ ...
Input table – W2
B1
B2
...
Bd2
R^{i}_{W1}
$PCT_{W2}{}^{i+1}$
R^{i+1}_{W2}
R1 : B1<y
R2: B1≥y
....
R8: B1≥y ∧ B5<z ∧ ..
R^{i+1}_{W1}; R^{i+1}_{W2}
R^{i-1}_{W1} ; R^{i-1}_{W2}
Redescription set expansion
Unguided: $R^{i+1}_{W1} \times R^{i+1}_{W2}$
Guided: $R^{i}_{W1} \times R^{i+1}_{W2}$;
$R^{i+1}_{W1} \times R^{i}_{W2}$

Fig. 1: Illustration of rule, redescription construction and iterations

the newly created redescription (lines 3-21 in Algorithm 2). Five different arrays ($elFreq$, $attrFreq$, $redScoreEl$, $redScoreAt$, $redDstC$) are used to incrementally improve and add redescriptions to the redescription set. The element/attribute frequency arrays contain information about element/attribute occurrence in redescriptions from a redescription set. Redescription scores (line 9 in Algorithm 2) are computed as $redScoreEl(R) = \sum_{e \in supp(R)} (elFreq[e] - 1)$, $redScoreAt(R) = \sum_{a \in attr(R)} (attrFreq[a] - 1)$, $redDstC(R) = \sum_{e \in supp(R)} \delta_{0,elFreq[e]-1}$. The score of a new redescription (line 18 in Algorithm 2) is computed in the same way by using existing frequencies from the set. For a redescription R' such that $R_i = argmax_{R \in \mathcal{R}}\ score(R', R)$, where $score(R', R) = (\frac{1.0 - R'.elSc + 1.0 - R'.atrSc + R'.J + R'.eDC + R'.pVSc)}{5} - \frac{(1.0 - R.elSc + 1.0 - R.attrSc + R.J + R.eDC + R.pVSc)}{5})$ and the $score(R', R_i) > 0$, all arrays are updated so that the frequencies of elements described by R_i and attributes contained in its queries are decreased by one, while the frequencies of elements and attributes associated with R' are increased (line 19 in Algorithm 2). The p-value score (R.pVSc) is computed as:

$$R.pVSc = \begin{cases} \frac{log_{10}(R.pval)}{-17.0} & \text{if } R.pval < 10^{-17} \\ 1.0 & \text{if } R.pval \geq 10^{-17} \end{cases}$$

We linearise and normalize redescription p-values to obtain a score that is used as one criteria in the optimization process with the aim of describing subsets of elements with queries that are unlikely to be created easily by matching a pair of randomly constructed queries. The $R.eDC$ denotes the element exclusive coverage and is defined as the fraction of elements that are described only by redescription $R (R.eDC = \frac{redDstC(R)}{|E|})$. $R.elSc$ and $R.atSc$ are obtained as a fraction of element frequencies for elements in redescription support or attributes used in its queries to total frequency of all elements or attributes. The score is defined to construct a redescription set by adding redescriptions that describe elements with low frequency by using non frequent attributes (to disallow redundancy) and, at the same time, finds as accurate and significant redescriptions as possible. Redescriptions describing unexplored elements are rewarded in the process (since we would like to describe as much elements as possible given the redescription set size). The score can be extended by defining importance weights for each criteria, providing users with the possibility to fine tune the redescription set optimization process.

Element weighting has been used before in subgroup discovery [14,20] to model covering importance for elements. Our approach is similar but uses different weighting mechanism, adapts it to the redescription mining setting by combining element and attribute weights and incorporates it into the framework of iterative redescription set refinement in which some redescriptions can be replaced with more suitable candidates. The *exclusive coverage* has been used in the work from Knobe and Ho [18] as one criteria for extracting a set of patterns.

If redescription set size is automatically determined, for each $R \in \mathcal{R}$ the algorithm computes $sc(R) = min_{R''\in\mathcal{R},\ R''\neq R}|score(R, R'')|$ (lines 11 and 16 in Algorithm 2). Measures of difference in quality characteristics between a given redescription and its closest neighbour serve as a statistics used to determine which newly created redescription should be used to expand the redescription set (increase it in size). We compute the Tukey's range $[Q_1 - k\cdot(Q_3 - Q_1), Q_3 + k\cdot(Q_3 - Q_1)]$ with $k = 1.5$ and denote the upper boundary as *out*. For each newly created redescription R', we compute $sc(R') = min_{R''\in\mathcal{R}} score(R', R'')$. If the redescription set contains a redescription with preferred quality score compared to newly created redescription R', the $sc(R')$ will be negative, thus the produced redescription is not allowed to expand the redescription set. Alternatively, if a newly created redescription has a positive score difference when compared to every redescription currently found in the redescription set, we require its difference to be at least as great as the computed value *out* from the set of score differences for redescriptions contained in the redescription set ($sc(R') \geq out$). If this condition is satisfied, the redescription is added to the redescription set, thus increasing its size (line 16 in Algorithm 2). Redescription satisfying this strict criterion has an exceptional quality, compared to all other redescriptions in the set. This can occur, for instance, if a redescription with maximal accuracy is found that describes a part of element and attribute space not explored by any redescription from the redescription set. If the required condition is not satisfied, newly created redescription is used to optimize the redescription set, possibly replacing some existing member. We are very conservative in increasing redescription set size suggested by the user because small sets are easier to explore thus preferable in redescription mining setting [8]. The redescription set expansion follows the general algorithm structure defined in the work by Bringmann et. al.([4]), though instead of enumerating all patterns, we optimize the set by creating and discarding a large number of redescriptions at each iteration. This makes the proposed algorithm memory efficient and allows redescription set optimization to be performed very quickly.

The algorithm can use three types of logical operators (disjunction, conjunction and negation) where using disjunction operators increases redescription accuracy and support (lines 22-33 in Algorithm 2). For a redescription $R = (q_1, q_2)$, we find rules r that maximize:

1. $J(supp(q_1 \vee r)\setminus supp(R), supp(q_2)\setminus supp(R))$

Algorithm 2 MineRed

```
Input: $RW_1$, $RW_2$, expansion type, ConstSet, iteration number, opSet, elFreq, attrFreq, redScoreEl, redScoreAt
Output: A set of redescriptions $\mathcal{R}$
procedure MineRed
    expansionSet ← returnExpansionSet(expansionType, opSet, $RW_1$, $RW_2$,
                        $SRW_1$, $SRW_2$)
    for $R' \in expansionSet$ do
        if (|$\mathcal{R}$|<ConstSet.MaxRed AND Const.SetSize==Fixed) OR
            (|$\mathcal{R}$|<max(20,ConstSet.MinRed) AND Const.SetSize!=Fixed) then
            updateFrequencies(elFreq, attrFreq, R')
            $\mathcal{R} \leftarrow \mathcal{R} \cup R'$
            if (|$\mathcal{R}$|==ConstSet.MaxRed AND Const.SetSize==Fixed) OR
                (|$\mathcal{R}$| == max(20,ConstSet.MinRed) AND Const.SetSize!=Fixed) then
                for $R \in \mathcal{R}$ do
                    computeScores(elFreq,attrFreq, redScoreEl, redScoreAt, redDstC, R)
                if Const.SetSize!=Fixed then
                    Stats← computeStatistics($\mathcal{R}$)
        else if (|$\mathcal{R}$|==ConstSet.MaxRed AND Const.SetSize==Fixed) OR
                (|$\mathcal{R}$| ≥max(20,ConstSet.MinRed) AND Const.SetSize!=Fixed) then
            if Const.SetSize!=Fixed AND expand($\mathcal{R}$,$R'$)==TRUE then
                $\mathcal{R} \leftarrow \mathcal{R} \cup R'$
                updtFreqAndScores(elFreq, attrFreq, redScoreEl, redScoreAt,redDstC, R')
                Stats← computeStatistics($\mathcal{R}$)
                continue
            compScore(elFreq,attrFreq, redScoreEl, redScoreAt, redDstC,R')
            $R_b \leftarrow argmax_{R \in \mathcal{R}}\ score(R', R)$
            updtFreqAndScores(elFreq, attrFreq, redScoreEl, redScoreAt,redDstC, R', R)
            $\mathcal{R} \leftarrow \mathcal{R} \backslash R_b \cup R'$
    if ∨ ∈ opSet then
        for R∈ $\mathcal{R}$ do
            if expansionType==unguidedExpansion AND iteration==0 then
                $ind \leftarrow 0$
            else
                $ind \leftarrow newRuleIt$
            $r'_{W_1} \leftarrow argmax(R.maxRef(r), R.maxRef(\neg r),\ r \in RW_{1_{ind}} \cup SRW_1)$
            $R_{ref} \leftarrow (r'_{W_1} \vee R.rW_1 \times R.rW_2)$
            $r'_{W_2} \leftarrow argmax(R_{ref}.maxRef(r), R_{ref}.maxRef(\neg r),\ r \in RW_{2_{ind}} \cup SRW_2)$
            $R_{ref} \leftarrow (R_{ref}.rW_1 \times r'_{W_2} \vee R.rW_2)$
            updtFreqAndScores(elFreq, attrFreq, redScoreEl, redScoreAt,redDstC ,R, $R_{ref}$)
            $\mathcal{R} \leftarrow \mathcal{R} \backslash R \cup R_{ref}$
    return $\mathcal{R}$
```

2. $J(supp(q_1 \vee \neg r) \backslash supp(R), supp(q_2) \backslash supp(R))$
3. $J(supp(q_1) \backslash supp(R), supp(q_2 \vee r) \backslash supp(R))$
4. $J(supp(q_1) \backslash supp(R), supp(q_2 \vee \neg r) \backslash supp(R))$

The rule r is found so that it covers elements that are supported by q_2 but not by q_1 ($R.maxRef(r'),\ r' \in RW_1$) and vice versa.

3.2 Algorithm time complexity

We train one predictive clustering tree model and a set of weak PCT learners contained in the random forest. Work from Stojanova et. al [29] shows that predictive clustering tree construction has the worst time complexity of $O(z \cdot m \cdot |E|^2)$ to completely induce the tree, where m denotes the number of descriptive variables in a selected view and z the total number of internal nodes in the tree. The number of PCT models in a random forest is a constant defined by the user which makes the complexity of training all PCT models equal to $O(z \cdot m \cdot |E|^2)$.

The elements are stored in the HashSet and the HashMap data structure with open addressing which have the time complexity of $O(1)$ for add, remove, contains and size assuming the hash function behaves in a random enough manner (uniform hashing).

As described in our previous work ([22]), the initialization step has the complexity of $O(|E| \cdot (|V_1| + |V_2|))$, the PCT to rules transformation has the complexity of $O(z)$, creation of redescriptions $O(z^2 \cdot |E|)$ and $O(z \cdot d \cdot |E|)$ if we have a balanced tree, where d equals the tree depth, which is a constant. Updating the attribute and element frequency tables and the total redescription scores has the complexity of $O(|E| + d)$ in average case and $O(z^2 \cdot (|E| + d))$ in the worst case when the set size grows proportionally with the number of created redescriptions. The computation of rules containing negation and disjunction operators has a complexity of $O(z \cdot |E|)$.

The minimization procedure has the time complexity of $O(|\mathcal{R}| \cdot ((a + a') \cdot |E| + (a^3 + a'^3) \cdot |E|))$, where a, a' represent the number of attributes in redescription rules which are constrained with the tree depth d (or a constant multiple of d in case of rules containing disjunctions). Since we have a an expandable set of redescriptions, the greatest possible number of redescriptions is a multiple of z^2. Thus, the worst case time complexity of the minimization procedure is $O(z^2 \cdot d^3 \cdot |E|^2)$ or $O(z^2 \cdot |E|^2)$ since d is a constant. In practice, due to very strict constraints, the size of a redescription set is very close to user suggested value and can be considered a constant. Thus, the average time complexity is $O(d^3 \cdot |E|)$ or $O(|E|)$, since d is a constant.

The total algorithm average time complexity equals: $O(z \cdot (|V_1| + |V_2|) \cdot |E|^2 + z^2 \cdot |E|)$ while the worst time complexity, assuming inadequate hashing function and a large resulting redescription set, is $O(z \cdot (|V_1| + |V_2| + z) \cdot |E|^2)$.

Optimizations that could speed up computing redescriptions include Local Sensitive Hashing [5] and the use of rule indexing that allows combining only those rules certain to cross the user defined thresholds if redescription accuracy constraints are defined.

4 Algorithm evaluation and comparison

In this section, we evaluate different extensions of the CLUS-RM algorithm and compare redescriptions produced by our algorithm with the current state of the art algorithms ReReMi [11], Split trees and Layered trees [34]. The algorithms are compared on three datasets with different properties. Since the Split trees and the Layered trees algorithms do not work with data containing missing values, we make comparison analysis only with the ReReMi algorithm on the Country dataset. On this dataset, we evaluate redescription accuracy by using two different measures: the pessimistic Jaccard index and the query - non missing Jaccard index presented in our previous work [22]. We use the notation from Galbrun and Miettinen [8] to denote $E_{1,1} = supp(q_1) \cap supp(q_2)$, $E_{1,0} = supp(q_1) \backslash supp(q_2)$, $E_{0,1} = supp(q_2) \backslash supp(q_1)$, $E_{1,?} = supp(q_1) \cap missing(q_2)$, $E_{?,1} = missing(q_1) \cap$

$supp(q_2)$, where $R = (q_1, q_2)$ and $missing(q)$ represents a set of elements containing missing values for some attribute in q. Pessimistic Jaccard index is defined as: $J_{pes}(R) = \frac{|E_{1,1}|}{|E_{1,0}|+|E_{0,1}|+|E_{1,1}|+|E_{1,?}|+|E_{?,1}|+|E_{0,?}|+|E_{?,0}|+|E_{?,?}|}$ and the query-non missing Jaccard index as: $J_{qnm}(R) = \frac{|E_{1,1}|}{|E_{1,0}|+|E_{0,1}|+|E_{1,1}|+|E_{1,?}|+|E_{?,1}|}$. These two measures are used because they guarantee that each element found in redescription support has defined values for all attributes in a whole query, if only conjunction operators are used, or in a part of a query describing this element subset, if all operators are used. Query non-missing Jaccard is more optimistic than pessimistic Jaccard ($J_{pess} \leq J_{qnm}$), because it disregards elements having undefined value for both redescription queries. We used the Siren tool [9] to perform redescription mining with ReReMi, Split trees and Layered trees algorithms.

4.1 Evaluation data

Evaluations and comparisons are performed on three datasets with different characteristics.

- The Country dataset [30,32,13] describes 199 different countries in the year 2012. The dataset has two views, both containing numerical attributes with possible missing values. The first view contains 49 attributes with country information obtained from the World Bank. The second view contains 312 attributes obtained from the UNCTAD database representing the ratio of import and export of a commodity compared to total import or export of a country in the year 2012.
- The Bio dataset [23,17,11] describes 2575 geographical locations in Europe. The dataset contains information about climate conditions (48 numerical attributes) for a certain location and the information about the presence of mammal species (194 boolean attributes) on these locations. The climate condition attributes contain average, maximum, minimum temperature and average monthly precipitation.
- The DBLP dataset [6,11] contains information about authors of scientific papers (6455 authors in total). The first view describes the author-conference bipartite graph (304 boolean attributes) and the second view describes the co-authorship network (6455 boolean attributes). This dataset is very sparse which makes it hard to find highly accurate redescriptions.

4.2 Algorithm parameters

In this section we explain all the parameters and settings used to perform evaluations and comparisons with various redescription mining algorithms.

Table 4: Constraints on redescriptions used by all current RM algorithms.

Constraint	Value range	Usual method of selecting a value
minimal Jaccard index	$[0, 1]$	This parameter is obtained by experimentation. While the goal is to get as highly accurate redescriptions as possible, it is sometimes necessary to lower the initially set minimal Jaccard index to obtain redescriptions. In general, higher Jaccard index increases redescription accuracy but decreases diversity.
maximal p-value	$[0, 1]$	This parameter is usually set to one of the two values: 0.05 or 0.01 since it denotes the significance level used as a threshold to accept redescriptions.
minimal support	$[1, \|E\|]$	This parameter is usually set to values > 1, since describing only one entity is rarely interesting. Setting the threshold is domain specific. It depends on what kind of groups with respect to the size might be interesting to the domain expert. Redescriptions with very large support ($> 0.8 \cdot \|E\|$) usually have lower theoretical and empirical statistical significance since it is easier to obtain high accuracy by random sampling of queries with such a large support. Also, these queries contain large part of the value distribution for the attributes used in its queries thus random permutation of values between entities has smaller effect on the accuracy of such redescriptions.

For all algorithms, we used maximal p-value threshold of 0.01 (the strictest significance level). The minimal Jaccard index was set to 0.2 level for the DBLP dataset (based on results presented in [11], Table 6.1, p. 46), 0.6 level for the Bio dataset (based on results in [11], Table 7, p. 301) and 0.5 level for the Country dataset (obtained by experimentation). Minimal support was set to 10 elements for the DBLP (based on [11], p. 46) and the same value is used for the Bio dataset. Minimal support is set to 5 elements for the Country dataset, since this dataset contains substantially smaller amount of elements and redescriptions describing properties of 5 different countries still seem interesting.

The algorithm specific parameter values used to create redescriptions are listed below.

- CLUS-RM - allows specifying maximum support which was set to 5036 for the DBLP dataset, 2060 for the Bio dataset and 120 for the trade dataset. Since it is possible to obtain redescriptions that describe all elements in the dataset by using disjunction, conjunction and negation operators, we set the maximum support to disallow such redescriptions. We used the *average tree depth* 8 for all datasets (this effectively determines the maximal number of attributes occurring in produced rules). 120 iterations were performed on the Bio and the DBLP dataset and 800 iterations on the Country dataset. Larger number of iterations produces larger variety of redescriptions. Since the Country dataset is much smaller than the DBLP and the Bio dataset, we could run larger number of iterations with smaller execution times than those obtained on the DBLP and the Bio dataset. We used regression trees in all experiments with random forest containing 50 trees (our estimate is that middle-sized workstations and smaller servers are already capable of running 50 threads in parallel, thus we used maximally 50 trees in the forest).
- ReReMi - we used *LHS/RHS max number of variables* = 15, *min contribution* = 3, *min uncovered* = 200. For the DBLP dataset we set *max number of pairs* = 1000, *Batch output* = 10 and *Batch capacity* = 50, for the Bio dataset we used *max number of pairs = 200. Batch output* and *Batch capacity* parameters were at their default values 1 and 4 respectively. For the view containing numerical values we used the default values. We also created redescription sets with the ReReMi algorithm that used only conjunction and literal level negation operators by using equivalent values of other setup parameters as to construct sets that were generated by using all operators. On the Country dataset, we

used *max product buckets* = 200, *max number of pairs* = 1000, for the set in which we allowed using only logical conjunction operator and *max number of pairs* = 500 in case in which we mined redescriptions by using all logical operators.

– Split trees and Layered trees - we used *max rounds* = 1000, and *max tree depth* = 15.

Explanations of all parameters used for the ReReMi, Split trees and Layered trees algorithm can be seen on the web page of the tool Siren: `http://siren.gforge.inria.fr/_static/miner_confdef.xml`. Values for parameters *min contribution* and *min uncovered* were set after discussion with the authors of the tool, parameters specifying maximal number of attributes, bathc output, capacity and maximal number of pairs were increased compared to default values to obtain larger number and more accurate redescriptions. After obtaining redescriptions with the ReReMi, Split trees and Layered trees algorithm, we used the *Filter redundant redescriptions* option to remove duplicate and redundant redescriptions with the *max overlap option* equal to 0.99.

For the evaluation of the CLUS-RM algorithm extensions, we use the same algorithm parameters as specified earlier. The exception is the number of iterations for the DBLP dataset which is set to 40. Also, we optimize a set containing 200 redescriptions.

4.3 Evaluating CLUS-RM extensions

In this section, we evaluate the effects of (a) using random forest based augmentation and (b) redescription set creation of user suggested size without specifying redescription accuracy constraints. First, we create a redescription set by using CLUS-RM with one PCT by using parameters specified in Section 4.2. Next, we use random forest based augmentation, with identical parameters and 50 trees in the forest, to optimize the set of the same size. The final experiment uses random forest based augmentation with identical parameters as before but without specifying redescription accuracy constraints and by allowing set expansion. In all experiments, we fix one random initialization for the initial step and use it to obtain all redescription sets. Also, random seeds of PCTs and the random forest is preserved between the experiments. In this way, we can explore the effects of different modifications made to the original CLUS-RM algorithm.
The experimental results related to the described extensions are presented in Figures 2,3,4 and 5.

4.3.1 Effects of using random forest based augmentation

The evaluation on the Country data, presented in Figure 2, reveals that using random forest based augmentation in fact decreases redescription set accuracy. The Country dataset contains a small number of elements, thus the algorithm manages to create very optimized set by using only one PCT. The large number of additional, diverse, redescriptions created with the random forest of PCTs is used to describe elements that are not described often in the set by using different subsets of attributes. This is reflected by lower average element and attribute

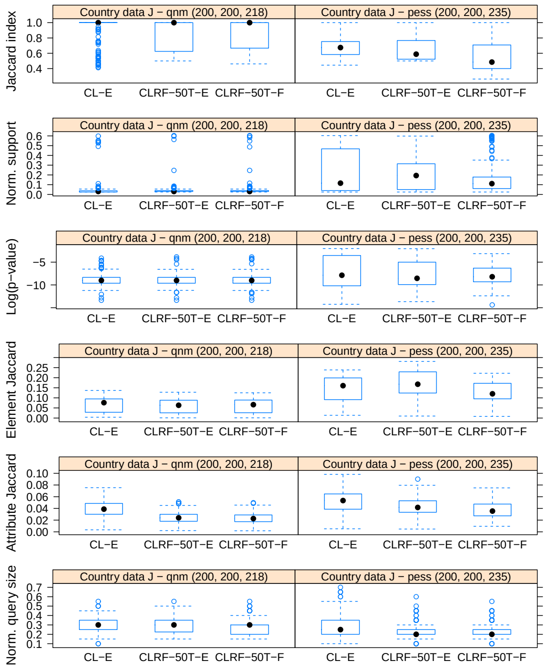


Fig. 2: Comparisons of redescription sets of size 200 produced by CLUS-RM using one PCT (CL-E), a PCT and a random forest containing 50 PCTs (CLRF-50T-E), and the CLUS-RM with flexible set size without specifying redescription accuracy using a PCT and a random forest containing 50 PCTs (CLRF-50T-F). The comparison is performed on the Country data.

Jaccard index in the set produced with the algorithm using random forest based augmentation. The algorithm iteratively describes the fraction of elements that can be described very accurately until the occurrence frequency of these elements in redescriptions does not become to high. When this happens, a number of accurate redescriptions are replaced with redescriptions that increase diversity but have lower accuracy. According to the one - tailed Mann - Whitney U test of statistical significance, the redescription set produced by using random forest tends to contain redescriptions with smaller average element (significant with $p = 0.01381$) and attribute Jaccard index (significant with $p < 2.2 \cdot 10^{-17}$) when query non-missing Jaccard was used and with smaller average attribute Jaccard index (significant with $p = 1.6 \cdot 10^{-9}$) when pessimistic Jaccard was used. Additional benefit of using random forest based augmentation on this set is that the produced set tends to have smaller query size in redescriptions (significant with $p = 0.01677$) when query non-missing Jaccard was used and (significant with $p = 8.4 \cdot 10^{-14}$) when pessimistic Jaccard was used.

To show that using random forest also increases the number of highly accurate redescriptions, we perform an additional experiment on the Country data by

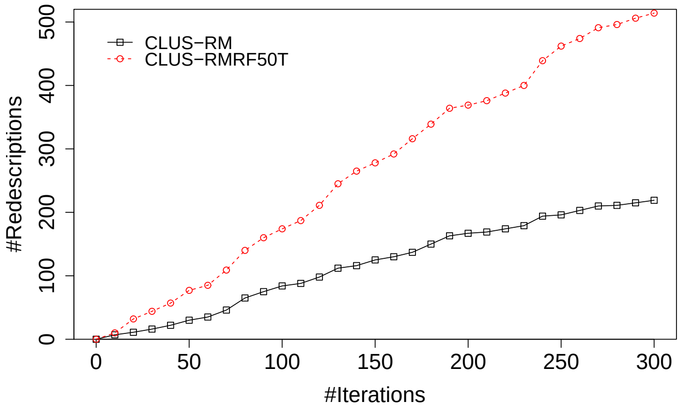


Fig. 3: Comparison of number of produced redescriptions with accuracy ≥ 0.9 by the CLUS-RM and the augmented variant with random forest containing 50 PCTs.

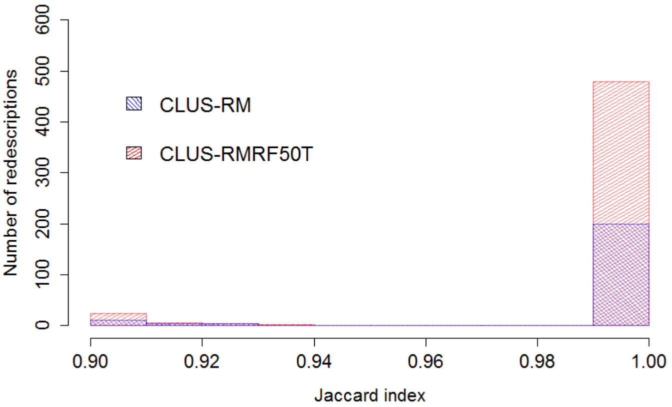


Fig. 4: Redescription accuracy distribution comparison between a set created by CLUS-RM and the augmented variant with random forest with 50 PCTs.

using query non-missing Jaccard index as accuracy measure. In this experiment, we set very strict accuracy threshold (minimal Jaccard index ≥ 0.9) required for redescription to be considered as a candidate to optimize redescription set. We return all distinct, highly accurate redescriptions created by CLUS-RM in 300 algorithm iterations by using one PCT and a PCT with random forest of PCTs. The change of number of highly accurate redescriptions through iterations are

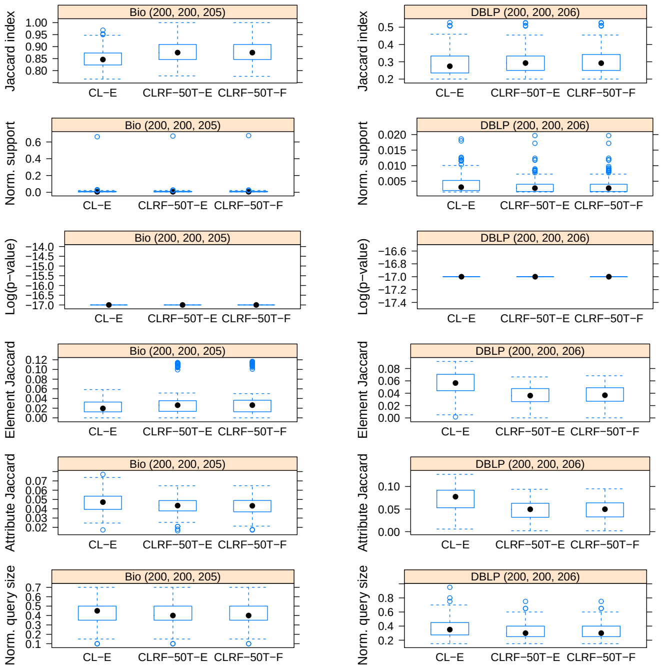


Fig. 5: Comparisons of redescription sets of size 200 produced by CLUS-RM using one PCT (CL-E), a PCT and a random forest containing 50 PCTs (CLRF-50T-E), and the CLUS-RM with flexible set size without specifying redescription accuracy using a PCT and a random forest containing 50 PCTs (CLRF-50T-F). The comparison is performed on the Bio data (left) and the DBLP data (right).

presented in Figure 3 and the comparative histogram displaying redescription accuracy distribution is shown in Figure 4.

The results presented in Figures 3 and 4 demonstrate the superior number of highly accurate redescriptions created by CLUS-RM algorithm augmented with random forest containing 50 PCTs. The difference between the number of generated highly accurate redescriptions increases at each algorithm iteration.

Using random forest based augmentation to create redescriptions on the Bio dataset, presented in Figure 5, allows creating redescription set containing redescriptions that tend to have higher accuracy ($p = 3.96 \cdot 10^{-11}$) than those produced when only one PCT is used. It also tends to have smaller average attribute Jaccard index ($p = 5.1 \cdot 10^{-5}$).

On the DBLP dataset (see Figure 5), using random forest allows creating redescription set that tends to contain redescriptions with higher accuracy ($p = 0.01088$) than those produced when only one PCT is used. Average element and attribute Jaccard index tend to be lower in the redescription set produced by using random forest, both with $p < 2.2 \cdot 10^{-16}$. Redescription query size tends to be lower in the redescription set produced by using random forest ($p = 0.04614$).

The experimental results presented in Figures 2 and 5 suggest that using random forest in CLUS-RM allows creating redescription sets with significantly lower average attribute Jaccard index on all used datasets which is important for exploring different associations. The same experiments suggest that it often results in obtaining significantly smaller redescription queries which is important for redescription understandability. The results presented in Figures 3, 4 and 5 show that it significantly increases redescription accuracy.

Since our method optimizes multiple objective criteria, it is not always possible to obtain domination even when the number of highly accurate redescriptions increases. This is visible from the results presented in Figures 2, 3 and 4. Here we can see that although the method augmented with random forest produces significantly larger number of highly accurate redescriptions, the overall redescription accuracy in the optimized redescription set decreases. The reason for this is that the benefits of describing larger number of diverse elements from the dataset by using more diverse attributes using redescriptions with smaller queries outweighs the benefits of adding more accurate but redundant and more complex redescriptions. On the DBLP dataset, the augmented model outperforms the basic algorithm on all measures (though it has slightly smaller redescription support).

4.3.2 Effects of creating redescription set of variable size without specifying accuracy constraints

The automatic set expansion procedure used on the Country dataset (Figure 2) increased the size of redescription set from initial 200 to 218 when query non-missing Jaccard was used. 10 redescriptions out of 18 have the maximal accuracy 1.0. The difference in accuracy between the set obtained by setting the redescription accuracy threshold and the set obtained without setting the threshold is not statistically significant. When pessimistic Jaccard was used, the procedure increased the size of redescription set with 35 additional redescriptions. However, the trade-off between accuracy and diversity resulted in lowering redescription accuracy to reduce the element and attribute diversity.

The redescription accuracy in the set created without specifying accuracy constraints on the Bio dataset (Figure 5) is not significantly different from that created with the fixed accuracy threshold.

On the DBLP dataset (Figure 5), the difference in accuracy between redescription set created by specifying redescription accuracy threshold and the set created without specifying this threshold is not statistically significant.

The results on all datasets, with the exception of using pessimistic Jaccard on the Country dataset, demonstrate that no significant drop in redescription accuracy occurs when redescription accuracy threshold is not set. The drop in accuracy occurs because of the trade-off between diversity, query size and accuracy.

4.4 Comparison with state of the art methods

Redescription mining algorithm comparison was mainly done in the literature by selecting and discussing properties of individual redescriptions. We try to make objective evaluation of redescription sets produced by different algorithms by using the same set of redescription constraints. Another condition we imposed is to have the same size of the final redescription sets. This is done by first finding redescription set with the ReReMi, Split trees, Layered trees algorithm, and then forcing the same size of the redescription set on the CLUS-RM, since it produces much more redescriptions than these algorithms.

The results are divided based on usage of logical operators. In the first experiment we allow using disjunctions, conjunctions, negations (DCN) and in the second experiment only conjunctions and negations (CN). For the generated redescription sets, we plot comparative boxplots for the Jaccard index, the log_{10} of the $p - value$, the average element and attribute Jaccard index and the redescription query size. We also compute the Man-Whitney U test to assess the statistical significance of the difference in algorithm performance. We only analyse redescription sets containing at least 10 redescriptions satisfying constraints. As a consequence, we can not make comparisons with Split trees and Layered trees algorithm in the CN mode on the DBLP and Bio dataset, and with Layered trees algorithm on the DBLP dataset in the DCN mode. We perform comparisons on the Country data only with ReReMi algorithm - by using only conjunction operator in CN mode. Other algorithms can not work on data containing missing values.

4.4.1 Comparison on the Country dataset

Comparative results on the Country dataset that contains missing values are obtained by optimizing J_{pes} with ReReMi algorithm and recalculating the score for each redescription to J_{qnm}. An optimized redescription set with the CLUS-RM was also created by using the pessimistic Jaccard index as one of the optimization criteria. The resulting sets are compared based on several quality criteria.

The results in Figure 6 show that the redescription set produced by our approach has higher median for redescription accuracy when all operators are used and query non-missing Jaccard is used to evaluate redescription accuracy. A slightly broader distribution is a result of redescription diversification. The Mann-Whitney U test of statistical significance shows that the set produced with CLUS-RM tends to contain more accurate redescriptions (significant with $p = 4.545 \cdot 10^{-5}$), it also tends to contain more significant redescriptions (significant with $p = 2 \cdot 10^{-9}$). The redescription set produced by the CLUS-RM tends to contain redescriptions with lower average element and attribute Jaccard index and with smaller query size (significant with $p < 2.2 \cdot 10^{-16}$). Redescriptions in the set produced by the CLUS-RM tend to contain redescriptions with smaller support (significant with $p < 2.2 \cdot 10^{-16}$). When only conjunction operators are allowed and query non-missing Jaccard is used, all measured quality criteria, except the average element Jaccard index, show that redescription set produced by CLUS-RM has significant advantage over ReReMi produced set. p-values obtained with one - tailed Mann-Whitney U test, presented in redescription quality criteria order as in Figure 6, are $p_J = 1.9 \cdot 10^{-7}$, $p_{supp} = 9.9 \cdot 10^{-5}$, $p_{pVal} = 1.3 \cdot 10^{-6}$, $p_{EJ} = 0.795$, $p_{AJ} = 9.3 \cdot 10^{-15}$, $p_{size} = 0.0007$. Redescription set (DCN) produced by ReReMi has the element coverage (EC) 1.0, the attribute coverage (AC) 0.35 and the set (CN) has $EC = 0.53$, $AC = 0.36$. The redescription set (DCN) produced by CLUS-RM has $EC = 0.99$ and $AC = 0.59$ and the set (CN) has $EC = 0.56$, $AC = 0.36$.

When pessimistic Jaccard index is used (Figure 7) to evaluate redescription accuracy, the redescription set created by CLUS-RM contains redescriptions that tend to have lower accuracy compared to ReReMi algorithm ($p = 0.01559$),

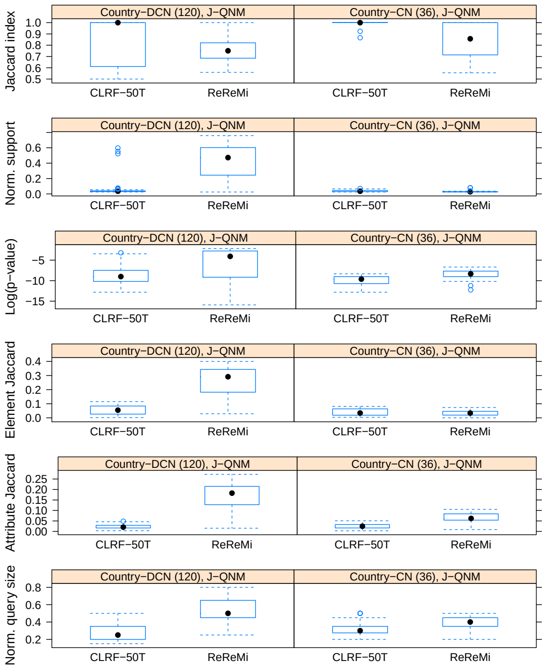


Fig. 6: Comparison of redescription sets produced by CLUS-RM and ReReMi algorithms on the Country dataset. Redescription accuracy is evaluated with the query non-missing Jaccard index.

they tend to have smaller support ($p = 8.84 \cdot 10^{-16}$), lower redescription p-value ($p = 3.88 \cdot 10^{-07}$), lower average element ($p = 2.076 \cdot 10^{-14}$) and attribute ($p < 2.2 \cdot 10^{-16}$) Jaccard index and smaller redescription query size ($p < 2.2 \cdot 10^{-16}$). When only conjunction operators are used, the CLUS-RM produced set contains redescriptions that tend to have higher redescription accuracy ($p = 7.873 \cdot 10^{-7}$), larger redescription support ($p = 0.00074$), lower redescription p-value ($p = 4.956 \cdot 10^{-6}$), lower average attribute Jaccard index ($p = 0.02652$) and smaller redescription query size ($p = 0.001727$). Redescription set created by CLUS-RM by optimizing pessimistic Jaccard index has $EC = 1.0$ and $AC = 0.5$ in the DCN mode, and $EC = 0.39$ and $AC = 0.31$ in the CN mode. The element coverage is slightly lower compared to ReReMi produced redescription set in the CN mode while the attribute coverage is comparable in CN mode and higher for CLUS-RM in the DCN mode. While ReReMi returned 2 redescriptions with $J_{pess} = 1.0$, CLUS-RM produced redescription set with 15 such redescriptions.

4.4.2 Comparison on the Bio dataset

Comparison results of redescription sets produced by CLUS-RM and ReReMi algorithm on the Bio dataset are presented in Figure 8.

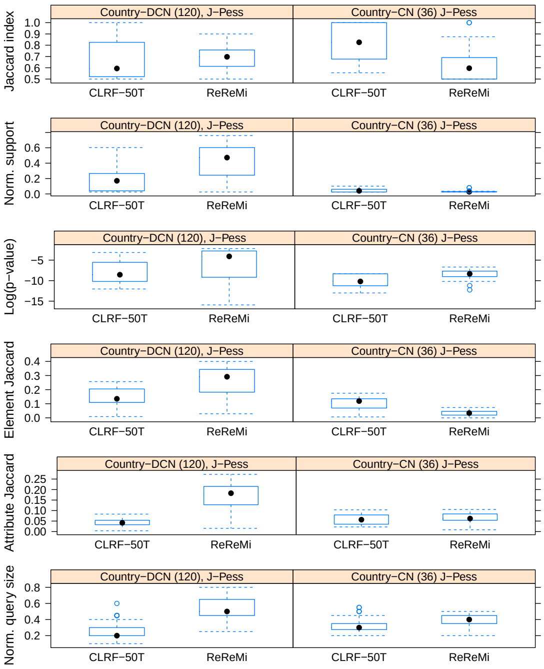


Fig. 7: Comparison of redescription sets produced by CLUS-RM and ReReMi algorithms on the Country dataset. Redescription accuracy is evaluated with the pessimistic Jaccard index.

The comparison results on the Bio dataset suggest that redescription set created by CLUS-RM contains redescriptions that tend to have higher accuracy compared to ReReMi algorithm ($p = 1.599 \cdot 10^{-5}$) when all logical operators are used to create redescriptions. They also tend to have lower redescription p-values ($p = 2.052 \cdot 10^{-9}$), lower average element ($p < 2.2 \cdot 10^{-16}$) and attribute ($p = 7.24 \cdot 10^{-8}$) Jaccard index as well as smaller redescription query size ($p = 0.03107$). However, they also tend to have smaller redescription support ($p < 2.2 \cdot 10^{-16}$). Our approach finds many redescriptions closer to minimal support boundary on the Bio dataset which complements ReReMi redescriptions that have a drift towards redescriptions with maximum support as reported in the work from Galbrun ([11]). When only conjunction operators are used to create redescriptions, our approach created redescription set that tends to contain more accurate redescriptions ($p = 8.31 \cdot 10^{-15}$), lower average element ($p = 1.403 \cdot 10^{-15}$) and attribute ($p = 7.666 \cdot 10^{-16}$) Jaccard index and smaller redescription query size ($p = 6.156 \cdot 10^{-5}$). It also tends to contain redescriptions with smaller redescription support ($p = 1.71 \cdot 10^{-15}$). The redescription set produced with ReReMi algorithm has $EC = 1.0$ and $AC = 0.39$ in DCN mode while CLUS-RM produced redescription set has $EC = 0.93$ and $AC = 0.59$. In the CN mode, the ReReMi produced redescription set has $EC = 0.98$ and $AC = 0.37$ while CLUS-RM pro-

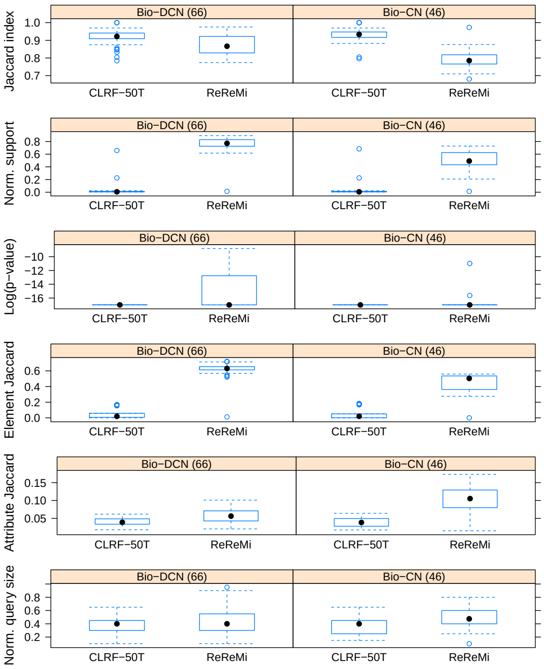


Fig. 8: Comparison of redescription sets produced by CLUS-RM and ReReMi algorithms on the Bio dataset.

duced redescription set has $EC = 0.95$ and $AC = 0.52$. The redescription sets produced with ReReMi and CLUS-RM have comparable element coverage but the redescription set produced with CLUS-RM has higer attribute coverage.

The comparison results of redescription sets produced by CLUS-RM, Split trees and Layered trees algorithm on the Bio dataset is available in Figure 9.

We perform comparisons of redescription sets created by CLUS-RM, Split trees and Layered trees algorithms by using all logical operators to construct redescriptions. The results suggest that the redescription set produced by CLUS-RM contains redescriptions that tend to have higher accuracy ($p = 2.738 \cdot 10^{-15}$), lower redescription p-value ($p = 0.02165$), lower average element ($p = 2.464 \cdot 10^{-11}$) and attribute ($p = 3.47 \cdot 10^{-16}$) Jaccard index and smaller redescription query size ($p = 0.02877$) compared to the redescription set created by Split trees algorithm. However, CLUS-RM produced redescription set also contains redescriptions that tend to have a smaller support ($p = 1.092 \cdot 10^{-14}$). The CLUS-RM produced redescription set has $EC = 0.75$ and $AC = 0.57$ while the Split trees algorithm produced redescription set with $EC = 0.98$ and $AC = 0.32$. The redescription set produced with CLUS-RM contains redescriptions that tend to have higher accuracy ($p = 2.978 \cdot 10^{-7}$), lower redescription p-value ($p = 0.04076$), lower average element ($p = 5.691 \cdot 10^{-8}$) and attribute ($p = 0.01856$) Jaccard index and smaller redescription query size ($p = 0.0006035$) compared to redescription set created by

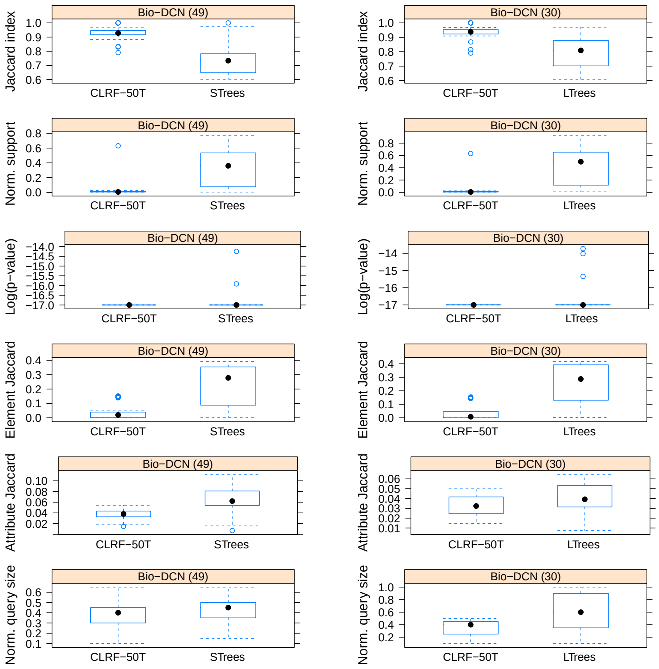


Fig. 9: Comparison of redescription set produced by CLUS-RM with the set produced by Split trees (left) and Layered trees (right) on the Bio dataset.

Layered trees algorithm. The redescription set produced by the CLUS-RM algorithm contains redescriptions that tend to have smaller support than redescriptions contained in the redescription set produced by the Layered tree algorithm ($p = 1.721 \cdot 10^{-8}$). The redescription set produced by CLUS-RM algorithm has $EC = 0.74$ and $AC = 0.45$ while the redescription set produced by Layered trees algorithm has $EC = 1.0$ and $AC = 0.53$.

Lower element coverage in CLUS-RM produced redescription set compared to redescription sets produced by Layered trees and Split trees algorithms is the consequence of a relatively small redescription set size: 49 and 30 redescriptions.

4.4.3 Comparison on the DBLP dataset

The comparison results of redescription sets produced by CLUS-RM, ReReMi and Split trees algorithm on the DBLP dataset is available in Figure 10. The DBLP dataset is very sparse and it is difficult to produce many highly accurate redescriptions. The results suggest that the redescription set produced by CLUS-RM algorithm contains redescriptions that tend to have higher accuracy ($p < 2.2 \cdot 10^{-16}$) and higher redescription support ($p = 1.23 \cdot 10^{-10}$), smaller redescription query size ($p = 0.005913$) compared to redescription set produced by the ReReMi algorithm when all logical operators were used to create redescriptions. Though, the redescription set produced by CLUS-RM contains redescriptions that tend to have higher redescription p-value ($p = 2.466 \cdot 10^{-14}$) and average element ($p < 2.2 \cdot 10^{-16}$) and attribute ($p < 2.2 \cdot 10^{-16}$) Jaccard index. The redescription set produced by CLUS-RM has $EC = 0.99$ and $AC = 0.06$ while the redescription set produced by ReReMi has $EC = 0.781$ and $AC = 0.326$. Since the authors form examples and attributes in this dataset, one potential explanation for smaller attribute coverage by CLUS-RM is that it concentrated the search to parts of DBLP network that can be described very accurately which constrained the diversity of authors and conferences occuring in redescription queries. This conclusion is also indicated by the higher attribute and element Jaccard index of the produced redescriptions which is visible on all performed experiments on this dataset. When only conjunction operators were used, the redescription set produced by CLUS-RM contains redescriptions that tend to have higher accuracy ($p < 2.2 \cdot 10^{-16}$), though they tend to have smaller support ($p = 2.133 \cdot 10^{-5}$), higher element ($p < 2.2 \cdot 10^{-16}$) and attribute ($p < 2.2 \cdot 10^{-16}$) Jaccard index and larger redescription query size ($p = 9.94 \cdot 10^{-15}$). The redescription set produced by the CLUS-RM algorithm has $EC = 0.044$ and $AC = 0.028$ while redescription set produced by ReReMi has $EC = 0.188 =$ and $AC = 0.044$.

The comparison with the redescription set produced by Split trees algorithm, available in Figure 10 (right), shows that redescription set produced by CLUS-RM algorithm contains redescriptions that do not have significant difference in accuracy, support and redescription p-value compared to redescriptions contained in the set created by Split trees algorithm, they tend to have redescriptions with smaller query size ($p = 1.145 \cdot 10^{-7}$) but they also tend to have larger average element ($p = 0.0003779$) and attribute ($p = 1.516 \cdot 10^{-8}$) Jaccard index. The redescription set produced by CLUS-RM has $EC = 0.067$ and $AC = 0.028$ while redescription set produced by Split trees has $EC = 0.085$ and $AC = 0.049$.

The results presented in this section lead us to conclude that the CLUS-RM algorithm outperforms other redescription mining approaches in the CN mode with respect to redescription accuracy. With the exception of the DBLP data, it also tends to have smaller average attribute Jaccard index, smaller redescription p-values and smaller query size. In the DCN mode, the approach outperformed other approaches in redescription accuracy, with the exception of the ReReMi algorithm when pessimistic Jaccard index is used to evaluate redescriptions on the Country dataset, and the Split trees on the DBLP dataset, where the difference in accuracy is not statistically significant.

The majority of produced redescriptions by the CLUS-RM contain conjunction and negation operators as opposed to redescriptions produced by other approaches that mostly contain disjunction operators. CLUS-RM uses disjunction operators sparingly by design because it requires redescriptions to have the accuracy larger than the minimal accuracy threshold in order to apply disjunction operator. This disallows CLUS-RM to create different disjunction - based redescriptions that describe unrelated parts of element space (and can have very high accuracy). Such redescriptions are found by ReReMi algorithm which is discussed by Galbrun [11]. This affects the number of highly accurate redescriptions produced by the CLUS-RM compared to other approach in DCN mode.

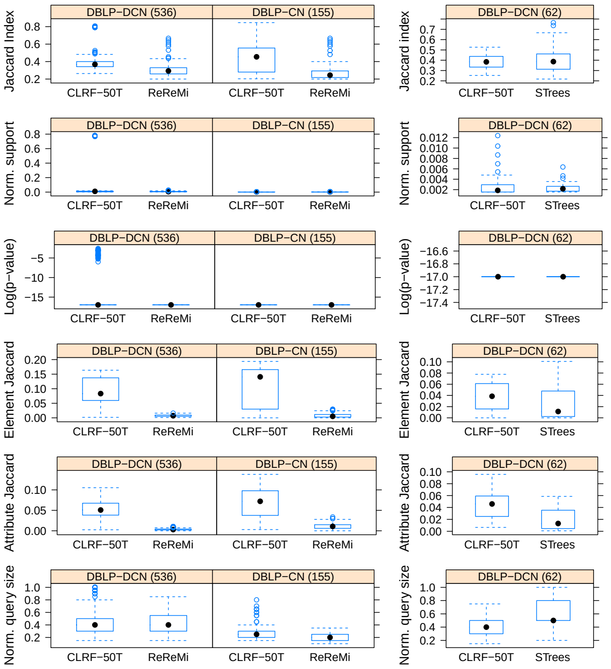


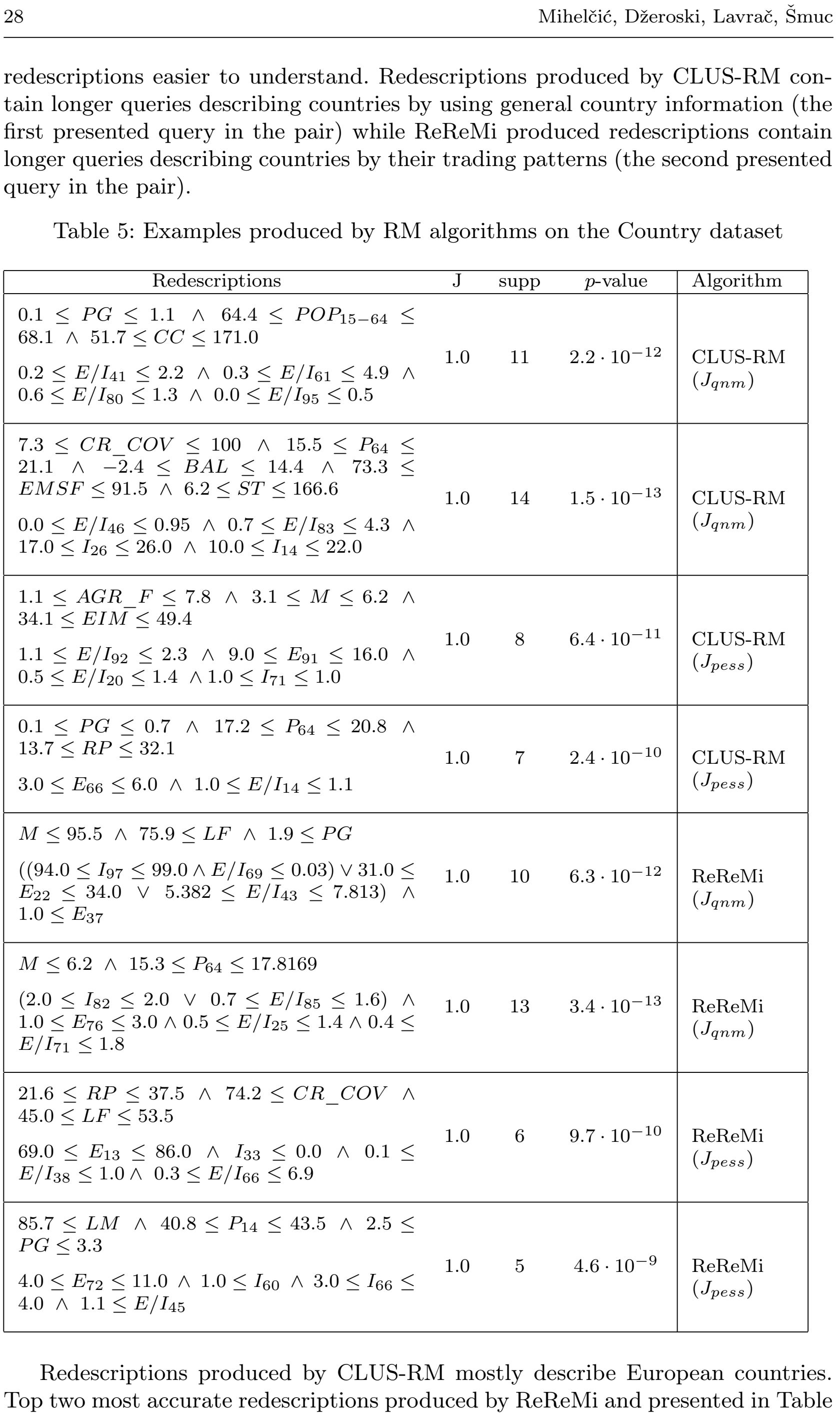
Fig. 10: Comparison of redescription sets produced by CLUS-RM and ReReMi algorithms (left), and the CLUS-RM and Split trees algorithm (right) on the DBLP dataset.

4.5 Redescription examples

In this section, we present top two redescriptions, by accuracy, found by each approach on the datasets used for evaluation. If there are multiple candidates with the same accuracy we choose redescriptions with shorter query size or smaller p-value. We compare these redescriptions by their structure and quality. The explanation of the meaning of these redescriptions along with a list describing all used attributes in redescription queries of these examples can be seen in Online resource 1.

4.5.1 Examples produced on the Country dataset

Redescriptions presented in Table 5 show that both CLUS-RM and ReReMi managed to find two redescriptions with maximal accuracy 1.0 with both Jaccard index variants. Redescriptions found by CLUS-RM have lower p-value and larger support. Structurally, redescriptions created with CLUS-RM contain only conjunction operators, while two redescriptions produced by ReReMi contain complex queries containing conjunction and disjunction operators. This makes CLUS-RM produced



redescriptions easier to understand. Redescriptions produced by CLUS-RM contain longer queries describing countries by using general country information (the first presented query in the pair) while ReReMi produced redescriptions contain longer queries describing countries by their trading patterns (the second presented query in the pair).

Table 5: Examples produced by RM algorithms on the Country dataset

Redescriptions	J	supp	p-value	Algorithm
$0.1 \leq PG \leq 1.1 \wedge 64.4 \leq POP_{15-64} \leq 68.1 \wedge 51.7 \leq CC \leq 171.0$ $0.2 \leq E/I_{41} \leq 2.2 \wedge 0.3 \leq E/I_{61} \leq 4.9 \wedge 0.6 \leq E/I_{80} \leq 1.3 \wedge 0.0 \leq E/I_{95} \leq 0.5$	1.0	11	$2.2 \cdot 10^{-12}$	CLUS-RM (J_{qnm})
$7.3 \leq CR\_COV \leq 100 \wedge 15.5 \leq P_{64} \leq 21.1 \wedge -2.4 \leq BAL \leq 14.4 \wedge 73.3 \leq EMSF \leq 91.5 \wedge 6.2 \leq ST \leq 166.6$ $0.0 \leq E/I_{46} \leq 0.95 \wedge 0.7 \leq E/I_{83} \leq 4.3 \wedge 17.0 \leq I_{26} \leq 26.0 \wedge 10.0 \leq I_{14} \leq 22.0$	1.0	14	$1.5 \cdot 10^{-13}$	CLUS-RM (J_{qnm})
$1.1 \leq AGR\_F \leq 7.8 \wedge 3.1 \leq M \leq 6.2 \wedge 34.1 \leq EIM \leq 49.4$ $1.1 \leq E/I_{92} \leq 2.3 \wedge 9.0 \leq E_{91} \leq 16.0 \wedge 0.5 \leq E/I_{20} \leq 1.4 \wedge 1.0 \leq I_{71} \leq 1.0$	1.0	8	$6.4 \cdot 10^{-11}$	CLUS-RM (J_{pess})
$0.1 \leq PG \leq 0.7 \wedge 17.2 \leq P_{64} \leq 20.8 \wedge 13.7 \leq RP \leq 32.1$ $3.0 \leq E_{66} \leq 6.0 \wedge 1.0 \leq E/I_{14} \leq 1.1$	1.0	7	$2.4 \cdot 10^{-10}$	CLUS-RM (J_{pess})
$M \leq 95.5 \wedge 75.9 \leq LF \wedge 1.9 \leq PG$ $((94.0 \leq I_{97} \leq 99.0 \wedge E/I_{69} \leq 0.03) \vee 31.0 \leq E_{22} \leq 34.0 \vee 5.382 \leq E/I_{43} \leq 7.813) \wedge 1.0 \leq E_{37}$	1.0	10	$6.3 \cdot 10^{-12}$	ReReMi (J_{qnm})
$M \leq 6.2 \wedge 15.3 \leq P_{64} \leq 17.8169$ $(2.0 \leq I_{82} \leq 2.0 \vee 0.7 \leq E/I_{85} \leq 1.6) \wedge 1.0 \leq E_{76} \leq 3.0 \wedge 0.5 \leq E/I_{25} \leq 1.4 \wedge 0.4 \leq E/I_{71} \leq 1.8$	1.0	13	$3.4 \cdot 10^{-13}$	ReReMi (J_{qnm})
$21.6 \leq RP \leq 37.5 \wedge 74.2 \leq CR\_COV \wedge 45.0 \leq LF \leq 53.5$ $69.0 \leq E_{13} \leq 86.0 \wedge I_{33} \leq 0.0 \wedge 0.1 \leq E/I_{38} \leq 1.0 \wedge 0.3 \leq E/I_{66} \leq 6.9$	1.0	6	$9.7 \cdot 10^{-10}$	ReReMi (J_{pess})
$85.7 \leq LM \wedge 40.8 \leq P_{14} \leq 43.5 \wedge 2.5 \leq PG \leq 3.3$ $4.0 \leq E_{72} \leq 11.0 \wedge 1.0 \leq I_{60} \wedge 3.0 \leq I_{66} \leq 4.0 \wedge 1.1 \leq E/I_{45}$	1.0	5	$4.6 \cdot 10^{-9}$	ReReMi (J_{pess})

Redescriptions produced by CLUS-RM mostly describe European countries. Top two most accurate redescriptions produced by ReReMi and presented in Table

5 are not as homogeneous as redescriptions produced by CLUS-RM with respect to location of described countries. They describe countries located in Africa, Asia and Europe. A detailed description of the meaning of these redescriptions along with the interpretation and confirmations from domain knowledge can be seen in Section S2.1 of the Online resource 1.

4.5.2 Examples produced by RM algorithms on the Bio dataset

Table 6 contains top two redescriptions (by accuracy) produced by each approach on the Bio dataset.

The exact locations used as examples in this dataset can be seen in [11], p. 50, Figure 6.1. On this dataset, we provide one additional redescription created by CLUS-RM as example of a redescription with large support. This redescription contains disjunction and negation operators. However, the top two redescriptions created by CLUS-RM contain only conjunction and negation operators. Presented redescriptions created by ReReMi and Layered trees contain all three logical operators and redescription examples created by Split trees algorithm contain conjunctions and disjunctions.

Redescriptions, presented in Table 6, with large support (> 1000 locations) are very uninformative because they contain negations of mammal species inhabiting geographical locations. All four algorithms discovered redescription describing habitats in Europe of the Polar Bear but use temperature in different months to provide information about the weather conditions on these locations. A detailed description of the meaning of these redescriptions along with the interpretation and confirmations from domain knowledge can be seen in Section S2.2 of the Online resource 1.

4.5.3 Examples produced by RM algorithms on the DBLP dataset

Table 7 contains top two redescriptions (by accuracy) produced by each approach on the DBLP dataset.

Structurally, CLUS-RM and ReReMi produced the most understandable redescriptions containing only conjunction and (CLUR-RM) negation operators. Split trees and Layered trees use all operators to create redescriptions which requires analysing queries in parts to understand the relationship of parts of a query to the corresponding part of redescription support which it describes.

A detailed description of the meaning of these redescriptions along with the interpretation and confirmations from domain knowledge can be seen in Section S2.3 of the Online resource 1.

We can see from the example tables (Table 6, 7) that the most accurate redescriptions created by the approaches on the Bio and the DBLP dataset have large similarities. All four approaches found a set of locations corresponding to a habitat of a Polar Bear on the Bio dataset but used different climate indicators to describe the weather on these locations. All algorithms discovered very similar sets of co-authors using slightly different authors and conferences in redescription queries.

Table 6: Examples produced by RM algorithms on the Bio dataset

Redescriptions	J	supp	p-value	Algorithm
$\neg(-8.3 \leq t_4^{\sim})$ PB	0.95	36	0.0	CLUS-RM (Bio)
$-8.4 \leq t_{11}^{\sim} \leq -5.98 \ \wedge \ 6.6 \leq t_6^{\sim} \leq 11.1 \ \wedge \ 78.9 \leq p_7 \leq 104.2 \ \wedge \ 14.4 \leq t_7^{+} \leq 18.8$ $\neg M \ \wedge \ B \ \wedge \ EWV \ \wedge \ W \ \wedge \ LS \ \wedge \ NB$	1.0	15	0.0	CLUS-RM (Bio)
$4.7 \leq t_3^{+} \leq 19.8$ $\neg M \ \vee \ C$	0.88	1726	0.0	CLUS-RM (Bio)
$-11.9 \leq t_3^{+} \leq -7.3$ PB	0.97	36	0.0	ReReMi (Bio)
$((((-12.2 \leq t_1^{-} \ \vee t_7^{+} \leq 13.5 \vee -12.2 \leq t_2^{\sim} \leq -11.8 \ \vee \ 13.3 \leq t_8^{\sim} \leq 13.8 \ \vee \ -2.4 \leq t_{11}^{\sim} \leq -1.7 \vee -7.6 \leq t_{12}^{\sim} \leq -7.5) \ \wedge -12.6 \leq t_3^{-} \leq -11.0) \ \vee \ 1.2 \leq t_4^{+} \leq 1.2) \ \wedge \ -2.4 \leq t_3^{+} \leq -1.5) \vee -6.5 \leq t_2^{+} \leq -6.4 \vee -4.58 \leq t_4^{\sim} \leq -4.55 \ \vee \ 12.5 \leq t_6^{\sim} \leq 12.5$ $\neg \ GRBV \ \wedge \ \neg W$	0.98	2294	0.0	ReReMi (Bio)
$(64.8 \leq p_{10}^{\sim} \wedge p_8^{\sim} \leq 2.2) \vee (34.4 \leq p_4^{\sim} \wedge p_9^{\sim} \leq 14.9 \ \wedge \ 2.2 \leq p_8^{\sim})$ CSM	1.0	10	0.0	Spl. Trees (Bio)
$(-16.7 \leq t_3^{\sim} \ \wedge \ t_3^{\sim} \leq -11.2)$ PB	0.97	36	0.0	Spl. Trees (Bio)
$16.6 \leq t_7^{+} \ \vee \ (16.6 \leq t_7^{+} \ \wedge \ 10.8 \leq t_9^{+})$ $(LW \ \wedge \ \neg AF) \ \vee \ (LW \ \wedge \ AF \ \wedge \ EH) \ \vee \ (\neg LW \wedge \ \neg AF)$	0.97	2370	$9.5 \cdot 10^{-15}$	Lay. Trees (Bio)
$t_3^{+} \leq -7.0$ PB	0.95	36	0.0	Lay. Trees (Bio)

5 Conclusions

This work introduces a novel redescription mining algorithm which optimizes a redescription set of user suggested size. The algorithm is based on multi-target predictive clustering trees, which allows using element coverage by rules constructed on one view as targets for the construction of rules from the other view. One predictive clustering tree is used to create rules that are employed to guide the search, while additional random forest of predictive clustering trees is used to construct rules that increase redescription accuracy and diversity. Produced redescriptions

Table 7: Examples produced by RM algorithms on the DBLP dataset

Redescriptions	J	supp	p-value	Algorithm
$\neg HICSS \land ISWC \land \neg ICWS \land EC-Web$ $AM \land CS \land DO$	0.83	10	0.0	CLUS-RM (DBLP)
$SEBD \land SIGMOD \land LPNMR$ $TE \land GG$	0.67	10	0.0	CLUS-RM (DBLP)
$SEBD \land LPNMR \land SIGMOD$ $TE \land GT$	0.67	10	0.0	ReReMi (DBLP)
$EC-Web \land ISWC$ $DO \land SH$	0.65	11	0.0	ReReMi (DBLP)
$(ITCC \land ISWC \land \neg EC-Web) \lor (\neg HICSS \land ISWC \land EC-Web)$ CS	0.76	13	0.0	Spl. Trees (DBLP)
$(ICIP(1) \land OTMW \land \neg EC-Web) \lor (\neg HICSS \land ISWC \land EC-Web)$ SH	0.74	14	0.0	Spl. Trees (DBLP)
$ISWC \land \neg HICSS \land \neg SIGIR$ $(SH \land CS \land \neg AG) \lor (\neg SH \land YSA)$	0.85	11	0.0	Lay. Trees (DBLP)
$ITCC \land SEBD \land \neg WETICE$ $(GM \land \neg AMa) \lor (\neg GM \land EM))$	0.81	13	0.0	Lay. Trees (DBLP)

incrementally improve the redescription set by using a predefined set of criteria (the Jaccard index, the p-value, the element and the attribute Jaccard index and the exclusive coverage). The ability to construct many different redescriptions and use them to optimize a redescription set differentiates the approach from currently proposed solutions and enables removing some user-defined constraints from the redescription mining process which is a desirable property [11]. The most important constraint not required by our approach is the Jaccard index threshold which is often determined by experimentation. Moreover, our approach expands the redescription set in very conservative manner which reflects the goal to present accurate and understandable redescription set of a user suggested size. Using random forest as the augmentation model decreases attribute redundancy and increases overall redescription accuracy in the output redescription sets in majority of experiments. It also increases the number of produced highly accurate redescriptions.

The results of algorithm comparisons show that our approach outperforms other approaches with respect to redescription accuracy when disjunction operators are not used in redescription construction. When all operators are used, CLUS-RM outperforms other approaches in majority of comparisons with respect to redescription accuracy. The final redescription sets contain redescriptions with smaller support, though the overall element and attribute coverage is comparable to other approaches. In general, CLUS-RM creates many different redescriptions of various support which can be obtained by increasing minimal support constraint or increasing the redescription set size.

We have demonstrated the advantages of our approach over current state of the art methods and provided exhaustive analysis that shows it creates many complementary redescriptions to those produced by currently proposed approaches. The produced redescriptions by our approach are structurally different, mostly containing conjunction operators in redescription queries and using disjunction operators only to improve accuracy of those redescriptions with accuracy above some predefined threshold. This increases understandability and eliminates (if high enough threshold is defined) creation of redescriptions describing unrelated parts of element space. Finally, we show that among top two redescriptions by accuracy, our approach has comparable performance with respect to other approaches. Among the example redescriptions, several redescriptions produced by different approaches have a large similarity in described elements and contain related queries. This is especially visible on the DBLP and the Bio dataset.

Acknowledgement

The authors would like to acknowledge the European Commission's support through the MAESTRA project (Gr. no. 612944), the MULTIPLEX project (Gr.no. 317532), the InnoMol project (Gr. no. 316289), and support of the Croatian Science Foundation (Pr. no. 9623: Machine Learning Algorithms for Insightful Analysis of Complex Data Structures).

References

1. Agrawal, R., Imieliński, T., Swami, A. (1993) Mining Association Rules Between Sets of Items in Large Databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 207-216, Washington, D.C.
2. Bickel, S., Scheffer, T. (2004) Multi-View Clustering. In Proceedings of the 4th IEEE International Conference on Data Mining, 19-26, Washington.
3. Blockeel., H. (1998) Top-down Induction of First Order Logical Decision Trees. Phd thesis, Katholieke Universiteit Leuven, Department of Computer Science.
4. Bringmann, B., Zimmermann A. (2007) The Chosen Few: On Identifying Valuable Patterns. Proceedings of the 7th IEEE International Conference on Data Mining, 63-72, Omaha.
5. Cohen, E., Datar, M., Fujiwara, S., Gionis, A., Indyk, P., Motwani, R., Ullman, J., D., Yang, C. (2000) Finding interesting associations without support pruning. In ICDE, 489-499.
6. DBLP dataset. `http://dblp.uni-trier.de/db` (March 2010)
7. Galbrun, E., Kimmig, A. (2014) Finding relational redescriptions. Machine Learning, 225-248.
8. Galbrun, E., Miettinen, P. (2012) From black and white to full color: extending redescription mining outside the Boolean world. Statistical Analysis and Data Mining, 284-303.
9. Galbrun, E. and Miettinen, P. Siren (2012) An Interactive Tool for Mining and Visualizing Geospatial Redescriptions. KDD, 1544-1547.
10. Galbrun, E., Miettinen, P. (2012) A Case of Visual and Interactive Data Analysis: Geospatial Redescription Mining. Instant Interactive Data Mining Workshop @ ECML-PKDD.
11. Galbrun, E. (2013) Methods for Redescription mining. Phd thesis, University of Helsinki.
12. Gallo, A., Miettinen, P., Mannila, H. (2008) Finding Subgroups having Several Descriptions: Algorithms for Redescription Mining. In Proceedings of the SIAM International Conference on Data Mining, 334-345, Atlanta, Georgia.
13. Gamberger, D., Mihelčić, M., Lavrač, N., Multilayer Clustering (2014) A Discovery Experiment on Country Level Trading Data. In Proceedings of the 17th International Conference on Discovery Science, 87-98, Bled. Slovenia.

14. Gamberger, D., Lavrač N. (2002) Expert-Guided Subgroup Discovery: Methodology and Application. Journal of Artificial Intelligence Research, 17, pp. 501-527.

15. Giacometti, A., Li, D. H., Marcel, P., Soulet, A. (2014) 20 Years of Pattern Mining: A Bibliometric Survey. SIGKDD Explor. Newsl., 41-50.

16. Han, J., Cheng, H., Xin, D., Yan, X., Frequent Pattern Mining (2007) Current Status and Future Directions. Data Mining and Knowledge Discovery, 15 : 55-86.

17. Hijmans R.J., Cameron S., Parra L., Jones P., and Jarvis A. Very high resolution interpolated climate surfaces for global land areas. International Journal of Climatology, 25, pp. 1965-978. (2005) `www.worldclim.org`

18. Knobbe A. J., Ho E. K. Y., (2006) Pattern Teams, In Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, Berlin, Germany, 577-584.

19. Kocev, D., K., Vens, C., Struyf, J., Džeroski, S. (2013) Tree ensembles for predicting structured outputs. Pattern Recognition, 817-833.

20. Lavrač, N., Kavšek, B., Flach, P., Todorovski Lj. (2004) Subgroup Discovery with CN2-SD, J. Mach. Learn. Res., 5, pp. 153-188.

21. Mihelčić, M., Džeroski S., Lavrač N., Šmuc. T. (2015) Redescription mining with multilabel Predictive Clustering Trees. In Proceedings of the 4th workshop on New Frontiers in Mining Complex Patterns, 86-97, Porto, Portugal.

22. Mihelčić, M., Džeroski S., Lavrač N., Šmuc. T. (2015) Redescription Mining with Multitarget Predictive Clustering Trees (2015) In New Frontiers in Mining Complex Patterns - 4th International Workshop, NFMCP 2015, Held in Conjunction with ECML-PKDD 2015, Porto, Portugal, September 7, 2015, Revised Selected Papers, 9607:125-143.

23. Mitchell-Jones A.J., Amori G., Bogdanowicz W., Krystufe B., Reijnders P., Spitzenberger F., Stubbe M., Thissen J., Vohralik V., and Zima J.: The Atlas of European Mammals. Academic Press, London (1999) `www.european-mammals.org`

24. Mooney, C. H., Roddick, J. F (2013) Sequential Pattern Mining – Approaches and Algorithms. ACM Computing Surveys, 45(2), ACM.

25. Parida, L., Ramakrishnan, N. (2004) Redescription Mining: Structure Theory and Algorithms. In Proceedings of the 20th National Conference on Artificial Intelligence, 837-844, Pittsburgh, Pennsylvania.

26. Piccart, B. (2012) Algorithms for Multi-Target Learning. Phd thesis, Katholieke Universiteit Leuven.

27. Ramakrishnan, N., Kumar, D., Mishra, B., Potts, M., Helm, R. F. (2004) Turning CARTwheels: an alternating algorithm for mining redescriptions. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 266-275, Seattle, WA.

28. Skyrienė G., Paulauskas A. (2013) Distribution of invasive muskrats (Ondatra zibethicus) and impact on ecosystem. Ekologija 58(3).

29. Stojanova, D., Ceci, M., Appice, A., Džeroski, S. (2012) Network regression with predictive clustering trees. Data Mining and Knowledge Discovery 378-413.

30. UNCTAD database, `http://unctadstat.unctad.org/EN/`.

31. van Leeuwen M., Galbrun, E. (2015) Association Discovery in Two-View Data. IEEE Transactions on Knowledge and Data Engineering 27:3190-3202.

32. World Bank database, `http://data.worldbank.org/`.

33. Zaki, M. J., and Ramakrishnan, N. Reasoning about sets using redescription mining (2005) In Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 364-373, Chicago, Illinois.

34. Zinchenko, T., Redescription Mining Over non-Binary Data Sets Using Decision Trees (2014) Masters thesis, Universität des Saarlandes.

Chapter 5

Redescription Set Optimization

A common goal of redescription mining approaches is to generate redescription sets containing interpretable, highly accurate, significant redescriptions. However, all previously constructed approaches evaluated redescriptions individually without taking into account information about other redescriptions returned to the user. We developed a methodology that constructs redescription sets by taking into account various user-defined preferences on redescription quality based on individual redescription properties and the properties of the redescription set under construction.

In this chapter we describe the process of redescription set optimization. First, we introduce the basic concepts and techniques for multi-objective optimization. Next, we describe the developed techniques for redescription set optimization: *optimization by redescription exchange* and *optimization by redescription extraction.* Finally, we describe the required extensions allowing redescription set optimization of redescriptions produced in the process of constraint-based redescription mining.

5.1 Introduction to Multi-Objective Optimization

Optimization is the task of finding one or more solutions which minimize/maximize one or more specified objectives and which satisfy all constraints (if any are provided) [81].

A single-objective optimization problem involves a single objective function $f : \mathbb{R}^n \mapsto \mathbb{R},\ n \in \mathbb{N}$ and usually results in a single solution. A multi-objective optimization task considers several possibly conflicting objectives simultaneously $\{f_1, f_2,\ \dots\ f_k\},\ k \in \mathbb{N},\ f_i : \mathbb{R}^n \mapsto \mathbb{R}$. In this case, there are usually more solutions, which represent a trade-off between the conflicting objectives. All solutions are non-dominated meaning that no solution has preferred values for all the considered objectives. In practice, one usually chooses one solution with some preferred properties. Thus, multi-objective optimization mostly consists of two parts:

- an computational optimization task for finding solutions
- a decision-making task for choosing a preferred solution. This step usually requires preference information from the decision maker (DM).

The field of multi-objective optimization is closely related to the field of multi-criteria decision aid (MCDA) [81], [82], which considers decision problems with multiple conflicting criteria. Many real-world optimization and decision problems naturally require considering multiple, possibly conflicting criteria. Such decisions occur very often in our lives: choosing groceries at the store, deciding on the type of public transport, renting the appropriate apartment, buying the preferred car etc.

5.1.1 Basic concepts of multi-objective optimization

In this section we provide basic notation and definitions required for understanding the problem of multi-objective optimization. The notation used closely follows that from [81].

The multi-objective optimization problems considered in this thesis are of the form:

$$\begin{aligned}&\min \{f_1(\vec{x}), f_2(\vec{x}), \ \dots \ , f_k(\vec{x})\}\\ &\vec{x} \in S \subset \mathbb{R}^n\end{aligned}$$

where $f_i : \mathbb{R}^n \mapsto \mathbb{R}$ represent $k \geq 2$ possibly conflicting *objective functions* that need to be minimized simultaneously. The decision (variable) vectors $\vec{x} = (x_1, x_2, \ \dots \ , x_n)^\tau$ belong to the set $S \subset \mathbb{R}^n$, which is called a *feasible region. Objective vectors* are defined as $\overrightarrow{f(\vec{x})} = (f_1(\vec{x}), f_2(\vec{x}), \ \dots \ f_k(\vec{x}))^\tau$. A set $Z = \vec{f}(S)$ is called a *feasible objective region.*

Objective vectors are regarded as optimal if none of their components can be improved without the deterioration of at least one of the other components, that is $\vec{x'}$ is *Pareto optimal* if $\nexists \vec{x} \in S$ such that: a) $f_i(\vec{x}) \leq f_i(\vec{x'})$, $\forall i \in \{1, \ \dots \ , k\}$ and b) $\exists j \in \{1, \dots, k\}$ $f_j(\vec{x}) < f_j(\vec{x'})$. $\vec{x'} \in S$ is *weakly Pareto optimal* if $\nexists \vec{x} \in S$, such that: a) $f_i(\vec{x}) < f_i(\vec{x'}), \forall i \in \{1, \ \dots \ , k\}$. A set of Pareto optimal decision vectors is denoted $P(S)$ while the set of Pareto optimal objective vectors is denoted $P(Z)$. For a Pareto optimal decision vector $\vec{x''} \in P(S)$, the objective vector $f(\vec{x''}) \in P(Z)$. A set of weakly Pareto optimal decision vectors is denoted $WP(S)$, while the set of weakly Pareto optimal objective vectors is denoted $WP(Z)$.

To analyse the Pareto optimal set, it is often useful to compute the upper and lower bounds of this set. The lower bound, called the *ideal objective vector*, is defined as ($z^* =$ (min $f_1(\vec{x})$, min $f_2(\vec{x})$, $\dots$, min $f_k(\vec{x})$), $\vec{x} \in S$). A vector, strictly dominating the ideal vector, is called the *utopian vector* (z^{**}, where $z_i^{**} = z_i^* - \varepsilon$, $\forall i \in \{1, \dots, k\}$ and some small scalar $\varepsilon > 0$). The upper bound, called the *nadir vector*, is defined as $z^{nad} =$ (max $f_1(\vec{x})$, max $f_2(\vec{x})$, $\dots$, max $f_k(\vec{x})$), $\vec{x} \in P(S)$.

The ideal vector can be obtained by minimizing each objective separately. However, the nadir vector can not be obtained in the same way since maximizing each objective separately provides the worst possible solution. Thus, knowledge about the whole Pareto front is required to obtain the nadir vector.

An often used method to find the nadir vector is a payoff table (developed by Benayoun et al. [83]). Unfortunately, it is not a reliable method, which has been demonstrated in [84], [85]. The payoff table method provides an accurate vector only if multi-objective optimization problem contains only two objectives. Otherwise, it can over(under)estimate because of alternative optima, explained in [86]. Various heuristic (evolutionary) approaches exist that compute the approximations of the nadir vector (e.g., [87]–[89]).

Basic concepts related to multi-objective optimization (in case of two objective functions) are visualized in Figure 5.1.

Pareto optimal solutions of multi-objective optimization problems exist if the feasible region is non-empty, compact and all the objective functions are lower semicontinuous (as shown in [90]). The same is true if the feasible objective region is non-empty and compact.

Continuous multi-objective optimization problems typically have an infinite number of Pareto optimal solutions, unlike combinatorial multi-objective optimization problems [82] that have a finite but possibly very large number of Pareto optimal solutions. In general, a Pareto optimal set can be non-convex and disconnected, which causes difficulties in obtaining all possible solutions with some multi-objective optimization approaches [81].

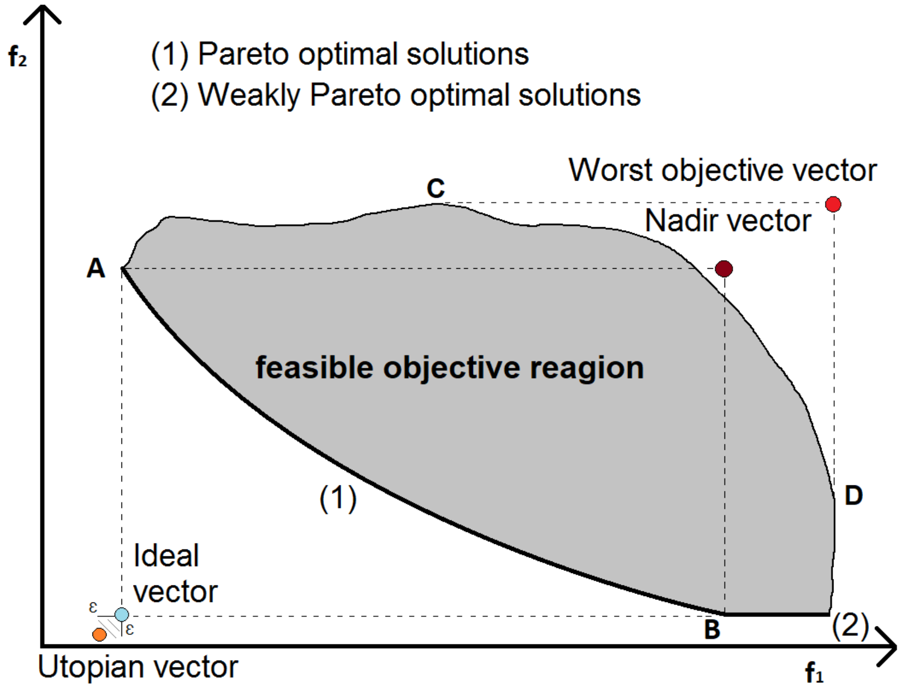

f₂
(1) Pareto optimal solutions
(2) Weakly Pareto optimal solutions
Worst objective vector
C
Nadir vector
A
feasible objective reagion
D
(1)
Ideal
vector
ε
ε
B
(2)
Utopian vector
f₁

Figure 5.1: Feasible objective region in case of two objective functions f_1 and f_2. The bold line denotes the weakly Pareto optimal set (2) and the Pareto optimal set (1).

5.1.2 Categorization of multi-objective optimization approaches

Depending on the type of interaction of the decision maker with the optimization process, multi-objective optimization approaches can be divided into: 1) *non-interactive* approaches, where the decision maker does not actively participate in the solution finding process but can define preference relations before or after the optimization process, 2) *interactive* approaches, where the decision maker actively participates in the process of finding solutions of a given multi-objective problem.

Non-interactive approaches can be divided into three overlapping groups of methods: 1) *no-preference* methods that do not use the information provided by the decision maker, 2) *a-priori* methods in which the decision maker first expresses his preferences on various optimization criteria and this information is used by the multi-objective optimization approaches to find satisfactory Pareto optimal solutions, and 3) *a-posteriori* methods in which the methods attempt to find all or multiple Pareto optimal solutions that are evaluated by the decision maker.

Since we use a non-interactive multi-objective optimization approach (the weighted sum method) to optimize sets of redescriptions, we briefly summarize the existing non-interactive multi-objective optimization approaches and place them in the corresponding category. For a detailed explanation and summary of advantages and disadvantages of each of these techniques we refer the interested reader to [81].

The *weighted sum* [91], [92] and the ε-constraint method [93], [94] are considered to be the basic methods for multi-objective optimization. By their construction, both methods can be a-priori or a-posteriori.

A-priori methods include: the *value function method* [95], the *lexicographic ordering* [96] and the *goal programming* [97], [98]. A-posteriori methods include: the *method of weighted metrics* [99], the *achievement scalarizing function approach* [100]. From the no-preference methods, the most important are: the *method of global criterion* [99], [101] and the *neutral compromise solution* [100].

In the continuation, we motivate our choice for using the weighted sum method in the redescription set optimization process (described in Sections 5.2 and 5.3), though using some other multi-objective approach in redescription set optimization or construction may also lead to interesting research directions.

5.1.3 Weighted-sum approach for multi-objective optimization

The main idea of the weighted-sum approach [91], [92] for multi-objective optimization is to optimize the weighted sum of the objective functions as a single-objective optimization function.

A general multi-objective problem:

$$\begin{aligned} &\min\ \{f_1(\vec{x}), f_2(\vec{x}),\ \dots\ , f_k(\vec{x})\} \\ &\vec{x} \in S \subset \mathbb{R}^n \end{aligned}$$

where $f_i : \mathbb{R}^n \mapsto \mathbb{R},\ S \subset \mathbb{R}^n,\ n \in \mathbb{N},\ k \geq 2$ is solved by the weighted-sum approach for multi-objective optimization [91], [92] by optimizing a single objective function:

$$\begin{aligned} &\min\ \textstyle\sum_{i=1}^{k} w_i \cdot f_i(\vec{x}) \\ &\vec{x} \in S \subset \mathbb{R}^n \end{aligned}$$

where weights $w_i \geq 0,\ i = 1, \dots, k$ such that usually (recommended) $\sum_{i=1}^{k} w_i = 1$. The solution of this method is in general weakly Pareto optimal and Pareto optimal if $w_i > 0,\ i = 1, \dots, k$ or unique solution exists [86].

The main advantages of this method are: a) it is very simple, b) for every given weight combination it finds one (weakly) Pareto optimal solution, c) it is computationally efficient (has equivalent complexity to solving a single-objective optimization), d) it can be used as an a-priori and an a-posteriori method.

The method is used as an a-priori method when the decision maker defines the weights corresponding to her preference and then finds one (weakly) Pareto optimal solution. On the other hand, a set of solutions can be generated by constantly changing weights and using a weighted-sum method to compute different solutions of the multi-objective problem. In the a-posteriori setting, the decision maker chooses the most suitable solution of the generated candidates.

The main drawback, in the view of multi-objective optimization, is that the weighted-sum method is not *complete*, meaning it can-not compute every Pareto optimal solution in case of a non-convex feasible objective region. The study of necessary conditions required for the weighted-sum method to obtain a Pareto optimal solution with a set of positive weights is reported in [102]. Ways of selecting input weights to obtain different Pareto optimal solutions is studied in [94], however it was shown in [103] that an evenly distributed set of weights does not necessarily produce an evenly distributed representation of the Pareto optimal set (even for convex problems). This may pose a problem for the decision maker in an a-priori and in an a-posteriori setting. If a multi-objective problem contains non-linear or correlated objective functions, choosing different weights for the weighted-sum method can produce unexpected solutions (small change in weights can produce very different solutions or an unexpected set of weights may result in the desired solutions) [104]–[106].

In Appendix A, we describe a constructive procedure that maps each redescription to a numerical vector, and show how to transform different redescription quality measures to equivalent measures operating on numerical vectors. For such measures and vectors, the original multi-objective optimization definition applies including all the properties of the weighted-sum method. Since there is a one-to-one mapping between the numerical vectors and redescriptions (under the assumption of no duplicate redescriptions—those with equal support sets and attribute sets) and the corresponding measures have equal values at each step of the optimization process, we can conclude that our optimization procedure finds Pareto optimal solution at each step of the redescription set optimization process. Thus, in the continuation, we use the originally defined functions operating on the domain containing redescriptions (defined in Chapter 3). Feasible region and the feasible objective region in our multi-objective optimization problem are two finite sets. Thus, both sets are non-convex which implies the weighted-sum method can not find all possible Pareto optimal solutions.

Multi-objective optimization is used at each step of the redescription set optimization process to find the redescription contained within existing set which will be replaced by the newly discovered redescription (in optimization by redescription exchange, see Section 5.2) or to find the appropriate redescription to be added to the redescription set under construction (in optimization by redescription extraction, see Section 5.3). Since the multi-objective optimization process is repeated many times, in redescription set optimization it is beneficial for it to be fully automated. The weighted-sum approach offers a fully automated, scalable solution that guarantees producing a (weakly) Pareto optimal solution at each step of redescription set optimization. It allows influencing the structure and different quality characteristics of the produced redescription sets by changing the weights used in the optimization process. Limitations concerning the choice of weights, optimization function and the control of the optimization process, described in the previous section, also apply in this scenario. Several experiments showing the influence of user-defined preference weights to the structure of the resulting redescription set are presented in publication [20]. In these experiments, changes of the weights used in the weighted-sum method to create the redescription sets are reflected on the properties of the constructed sets.

5.2 Optimization by Redescription Exchange

Optimization by redescription exchange [18], [19] is described in Algorithm 5.1 .

Algorithm 5.1: ReduceSetE

```
Input: The redescription set $R_{opt}$, redescriptions produced at iteration $i$ $\mathcal{R}_i$,
       Importance weight vector $\vec{w}$, Size of reduced set $n$, Set of constraints $C$
Output: An optimized set of redescriptions $\mathcal{R}_{opt}$

while $|\mathcal{R}_{opt}| < n$ do
    $\mathcal{R}_{opt} \leftarrow \mathcal{R}_{opt} \cup \{R_{new}\},\ R_{new} \in \mathcal{R}_i,\ R_{new} \notin \mathcal{R}_{opt},\ R_{new}.sat(C)$;
    $\mathcal{R}_i \leftarrow \mathcal{R}_i \setminus R$;
end
for $R_{new} \in \mathcal{R}_i$ do
    if $R_{new}.sat(C)$ then
        $R' \leftarrow argmax_{R \in \mathcal{R}_{opt}} f(R_{new}, \vec{w}) - f(R, \vec{w})$;
        if $max_{R \in \mathcal{R}_{opt}} f(R_{new}, \vec{w}) - f(R, \vec{w}) > 0$ then
            $R_{opt} \leftarrow R_{opt} \setminus \{R'\} \cup \{R\}$;
        end
    end
end
return $\mathcal{R}_{opt}$;
```

This algorithm works by incrementally exchanging redescriptions contained in the redescription set by newly produced redescriptions with superior properties with respect to predefined quality criteria. The redescription that is removed from the redescription set is chosen by computing $argmax_{R\in\mathcal{R}_{opt}} f(R,\vec{w}) - f(R_{new},\vec{w})$. The details of the optimization function f are provided in [19] and [18]. Here, we show a generalized version of the optimization by redescription exchange procedure with criteria importance weight vector $\vec{w}$. The notation $R_{new}.sat(C)$ denotes that the redescription R_{new} satisfies the user-defined quality constraints C. This algorithm can be run with different configurations of constraints ($C_i \in \mathcal{C}$) and weight parameters $\vec{w_i} \in \mathcal{W}$, however it requires storing $|\mathcal{C}| \cdot |\mathcal{W}|$ redescription sets in memory.

5.3 Optimization by Redescription Extraction

Optimization by redescription extraction (described in Algorithm 5.2) [20] works by storing all (or a large amount) of produced redescriptions into memory and uses these redescriptions to construct one or more reduced sets with different properties. Functions used in Algorithm 5.2 are explained in [20].

Algorithm 5.2: ReduceSet

```
Input: Redescription set containing all produced redescriptions $\mathcal{R}$, Importance
       weight matrix $\mathcal{W}$, Family of redescription constraints $\mathcal{C}$, Reduced set size $n$
Output: A family of optimized sets of redescriptions $\mathcal{R}_{Fopt}$

$[E_{ocur}, A_{ocur}] \leftarrow$ computeCoocurence($\mathcal{R}$);
for $w_i \in \mathcal{W}$ do
    for $C_i \in \mathcal{C}$ do
        $R_{first} \leftarrow$ findSpecificRed($\mathcal{R}, E_{cooc}, A_{cooc}, w_i, C_i$);
        $\mathcal{R}_{w_i,C_i} \leftarrow \mathcal{R}_{w_i,C_i} \cup R_{first}$;
        while $|R_{w_i,C_i}| < n$ do
            $R_{best} \leftarrow$ findBest($\mathcal{R}, \mathcal{R}_{w_i,C_i}, w_i, C_i$);
            $\mathcal{R}_{w_i,C_i} \leftarrow \mathcal{R}_{w_i,C_i} \cup R_{best}$;
        end
        $\mathcal{R}_{w_i,C_i} \leftarrow \mathcal{R}_{w_i,C_i} \cup R_{best}$;
    end
    $\mathcal{R}_{Fopt} \leftarrow \mathcal{R}_{Fopt} \cup \{\mathcal{R}_{w_i,C_i}\}$;
end
return $\mathcal{R}_{Fopt}$;
```

The main difference between the redescription set optimization by redescription extraction and redescription set optimization by redescription exchange is that the extraction procedure considers all available (produced) redescriptions at each iteration to determine the best match to be placed in the reduced sets, given the user-defined redescription constraints, criteria importance weights and redescriptions contained in the reduced redescription set. Given the properties of the weighted-sum multi-objective optimization procedure, the extraction process locates the Pareto optimal solution, that is the redescription that fits best with redescriptions already contained in the reduced set, given a predefined set of quality and preference criteria. In an idealized scenario where we could obtain all possible redescriptions, this procedure finds globally Pareto optimal redescription for a reduced redescription set at each iteration of the extraction process. Optimizing a submodular function that evaluates the optimal set arrangement out of all possible arrangements is a

NP-hard problem [107]. Determining the deviations from optimal solution theoretically or empirically is an interesting future work direction.

5.4 Redescription Set Optimization with User-Defined Constraints

Constraints as used in this thesis specify the attributes that must/should occur in redescription queries. Constraints are given as a sequence of sets $AC_1, AC_2, \ldots, AC_k$, where AC_i specifies one set of constraint attributes defined by the user.

Functions used in redescription set optimization were extended with the measure that evaluates the redescription agreement with user-defined attribute constraints (entity constraints can be added analogously) [22]. Depending on the mode of constraint-based redescription mining the optimization procedure:

a) disregards all redescriptions that do not fully satisfy at least one set of constraints imposed by the user,

b) disregards all redescriptions that do not satisfy at least some part of at least one set of constraints imposed by the user,

c) uses the additionally defined constraint-based measure to adequately increase the score of a redescription.

The constraint-based measure assesses the amount of imposed constraints satisfied by the query. Since in a general case, users can define a sequence of sets of constraints, the procedure selects the constraint set that is maximally satisfied by the redescription and assesses the amount of constraints satisfied from this set. The obtained score is combined with the score measuring the fraction of the query attributes satisfying any user-defined constraint (contained in any specified constraint set) to evaluate redescriptions with respect to predefined attribute constraints.

Different modes of constraint-based redescription mining and the optimization function in the case attribute constraints are imposed by the user are defined in [22].

5.5 Related Publication

Details of a framework for redescription set construction including descriptions of redescription set optimization by redescription extraction and providing various experiments showing properties of the produced redescription set are described in the following publication (included in this chapter):

M. Mihelčić, S. Džeroski, N. Lavrač, and T. Šmuc, “A framework for redescription set construction,” *Expert Systems with Applications*, vol. 68, pp. 196–215, 2017, ISSN: 0957-4174.

The author contributions are as follows. Matej Mihelčić introduced the idea of redescription set optimization by redescription extraction, devised and implemented the generalized redescription set construction procedure, conjunctive refinement procedure and the variability index. He performed all experiments, wrote the majority of the manuscript text and created all supplementary material documents. Tomislav Šmuc, Sašo Džeroski and Nada Lavrač contributed towards structuring, evaluating, correcting and writing the text of the manuscript.

A framework for redescription set construction

Matej Mihelčić[a,c,*], Sašo Džeroski[b,c], Nada Lavrač[b,c], Tomislav Šmuc[a]

[a] *Ruđer Bošković Institute, Bijenička cesta 54, 10000 Zagreb, Croatia*
[b] *Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia*
[c] *International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia*

This is a post-peer-review, pre-copy edit version of an article published in Journal of Experts Systems with Applications. The final authenticated version is available online at: https://doi.org/10.1016/j.eswa.2016.10.012.

Abstract

Redescription mining is a field of knowledge discovery that aims at finding different descriptions of similar subsets of instances in the data. These descriptions are represented as rules inferred from one or more disjoint sets of attributes, called views. As such, they support knowledge discovery process and help domain experts in formulating new hypotheses or constructing new knowledge bases and decision support systems. In contrast to previous approaches that typically create one smaller set of redescriptions satisfying a pre-defined set of constraints, we introduce a framework that creates large and heterogeneous redescription set from which user/expert can extract compact sets of differing properties, according to its own preferences. Construction of large and heterogeneous redescription set relies on CLUS-RM algorithm and a novel, conjunctive refinement procedure that facilitates generation of larger and more accurate redescription sets. The work also introduces the variability of redescription accuracy when missing values are present in the data, which significantly extends applicability of the method. Crucial part of the framework is the redescription set extraction based on heuristic multi-objective optimization procedure that allows user to define importance levels towards one or more redescription quality criteria. We provide both theoretical and empirical comparison of the novel framework against current state of the art redescription mining algorithms and show that it represents more efficient and versatile approach for mining redescriptions from data.

Keywords: knowledge discovery, redescription mining, predictive clustering trees, redescription set construction, scalarization, conjunctive refinement, redescription variability

1. Introduction

In many scientific fields, there is a growing need to understand measured or observed data, to find different regularities or anomalies, groups of instances (patterns) for which they occur and their descriptions in order to get an insight into the underlying phenomena.

This is addressed by redescription mining (Ramakrishnan et al., 2004), a type of knowledge discovery that aims to find different descriptions of similar sets of instances by using one, or more disjoint sets of descriptive attributes, called views. It is applicable in a variety of scientific fields like biology, economy, pharmacy, ecology, social science and other, where it is important to understand connections between different descriptors and to find regularities that are valid for different subsets of instances. Redescriptions are tuples of logical formulas which are called queries. Redescription $R_{ex} = (q_1, q_2)$ contains two queries:
q_1 : $(-1.8 \leq \tilde{t}_7 \leq 4.4 \land 12.1 \leq \tilde{p}_6 \leq 21.2)$
q_2 : Polarbear
The first query (q_1') describes a set of instances (geospatial locations) by using a set of attributes related to temperature (t) and precipitation (p) in a given month as first view (in the example average temperature in July and average precipitation in June). The second query (q_2') describes very similar set of locations by using a set of attributes specifying animal species inhabiting these locations as a second view (in this instance polar bear). Queries contain only conjunction logical operator, though the approach supports conjunction, negation and disjunction operators.

We first describe the fields of data mining and knowledge discovery closely related to redescription mining. Next, we describe recent research in redescription mining, relevant to the approach we propose. We then outline our approach positioned in the context of related work.

1.1. Fields related to redescription mining

Redescription mining is related to association rule mining (Agrawal et al., 1996; Hipp et al., 2000; Zhang & He, 2010), two-view data association discovery (van Leeuwen & Galbrun, 2015), clustering (Cox, 1957; Fisher, 1958; Ward, 1963; Jain et al., 1999; Xu & Tian, 2015) and

[*] Corresponding author. Tel. +385 (1) 456 1080
Email addresses: matej.mihelcic@irb.hr (Matej Mihelčić), saso.dzeroski@ijs.si (Sašo Džeroski), nada.lavrac@ijs.si (Nada Lavrač), tomislav.smuc@irb.hr (Tomislav Šmuc)

it's special form conceptual clustering (Michalski, 1980; Fisher, 1987), subgroup discovery (Klösgen, 1996; Wrobel, 1997; Novak et al., 2009; Herrera et al., 2010), emerging patterns (Dong & Li, 1999; Novak et al., 2009), contrast set mining (Bay & Pazzani, 2001; Novak et al., 2009) and exceptional model mining (Leman et al., 2008). Most important relations can be seen in Figure 1.

Association rule mining (Agrawal et al., 1996) is related to redescription mining in the aim to find queries describing similar sets of instances which reveal associations between attributes used in these queries. The main difference is that association rules produce one directional associations while redescription mining produces bi directional associations. Two-view data association discovery (van Leeuwen & Galbrun, 2015) aims at finding a small, non - redundant set of associations that provide insight in how two views are related. Produced associations are both uni and bi directional as opposed to redescription mining that only produces bi directional connections providing interesting descriptions of instances.

The main goal of clustering is to find groups of similar instances with respect to a set of attributes. However, it does not provide understandable and concise descriptions of these groups which are often complex and hard to find. This is resolved in conceptual clustering Michalski (1980); Fisher (1987) that finds clusters and concepts that describe them. Redescription mining shares this aim but requires each discovered cluster to be described by at least two concepts. Clustering is extended by multi-view (Bickel & Scheffer, 2004; Wang et al., 2013) and multi-layer clustering (Gamberger et al., 2014) to find groups of instances that are strongly connected across multiple views.

Subgroup discovery (Klösgen, 1996; Wrobel, 1997) differs from redescription mining in its goals. It finds queries describing groups of instances having unusual and interesting statistical properties on their target variable which are often unavailable in purely descriptive tasks. Exceptional model mining (Leman et al., 2008) extends subgroup discovery to more complex target concepts searching for subgroups such that a model trained on this subgroup is exceptional based on some property.

Emerging Patterns (Dong & Li, 1999) aim at finding itemsets that are statistically dependent on a specific target class while Contrast Set Mining (Bay & Pazzani, 2001) identifies monotone conjunctive queries that best discriminate between instances containing one target class from all other instances.

1.2. Related work in redescription mining

The field of redescription mining was introduced by Ramakrishnan et al. (2004), who present an algorithm to mine redescriptions based on decision trees, called CARTwheels. The algorithm works by building two decision trees (one for each view) that are joined in the leaves. Redescriptions are found by examining the paths from the root node of the first tree to the root node of the second. The algorithm uses multi class classification to guide the search between the two views. Other approaches to mine redescriptions include the one proposed by Zaki & Ramakrishnan (2005), which uses a lattice of closed descriptor sets to find redescriptions; the algorithm for mining exact and approximate redescriptions by Parida & Ramakrishnan (2005) that uses relaxation lattice, and the greedy and the MID algorithm based on frequent itemset mining by Gallo et al. (2008). All these approaches work only on Boolean data.

Galbrun & Miettinen (2012b) extend the greedy approach by Gallo et al. (2008) to work on numerical data. Redescription mining was extended by Galbrun & Kimmig (2013) to a relational and by Galbrun & Miettinen (2012a) to an interactive setting. Recently, two tree-based algorithms have been proposed by Zinchenko (2014), which explore the use of decision trees in a non-Boolean setting and present different methods of layer-by-layer tree construction, which make informed splits at each level of the tree. Mihelčić et al. (2015a,b) proposed a redescription mining algorithm based on multi-target predictive clustering trees (PCTs) (Blockeel & De Raedt, 1998; Kocev et al., 2013). This algorithm typically creates a large number of redescriptions by executing PCTs iteratively: it uses rules created for one view of attributes in one iteration, as target attributes for generating rules for the other view of attributes in the next iteration. A redescription set of a given size is improved over the iterations by introducing more suitable redescriptions which replace the ones that are inferior according to predefined quality criteria.

In this work, we introduce a redescription mining framework that allows creating multiple redescription sets of user defined size, based on user defined importance levels of one or more redescription quality criteria. The underlying redescription mining algorithm uses multi-target predictive clustering trees (Kocev et al., 2013) and allows the main steps of rule creation and redescription construction explained in (Mihelčić et al., 2015b). This is in contrast to current state of the art approaches that return all constructed redescriptions that satisfy accuracy and support constraints (Ramakrishnan et al., 2004; Zaki & Ramakrishnan, 2005; Parida & Ramakrishnan, 2005), a smaller number of accurate and significant redescriptions that satisfy support constraints (Galbrun & Miettinen, 2012b; Zinchenko, 2014; Gallo et al., 2008) or optimize one redescription set of user defined size (Mihelčić et al., 2015b). This algorithm supports a broader process which involves the creation and effective utilization of a possibly large redescription set.

From the expert systems perspective, the framework allows creating large and heterogeneous knowledge basis for use by the domain experts. It also allows fully automated construction of specific subsets of obtained knowledge based on predefined user-criteria. The system is modular and allows using the redescription set construction procedure as an independent querying system on the database created by merging multiple redescription sets produced by many different redescription mining approaches. Ob-

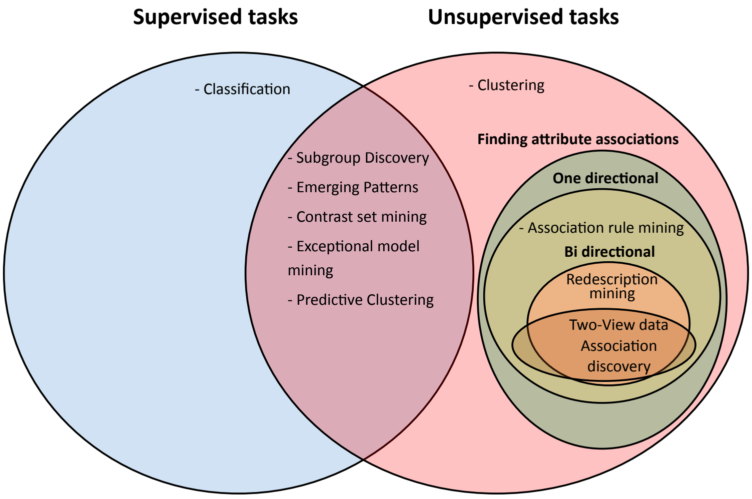


Figure 1: Relation between redescription mining and other related tasks.

tained knowledge can be used, for example, as a basis or complement in decision support systems.

The framework provides means to explore and compare multiple redescription sets, without the need to expensively experiment with tuning the parameters of the underlying redescription mining algorithm. This is achieved with (i) an efficient redescription mining algorithm with a new conjunctive refinement procedure, that produces large, heterogeneous and accurate redescription sets and (ii) redescription set construction procedure that produces one or more reduced redescription sets tailored to specific user preferences in a multi-objective optimization manner.

After introducing the necessary notation in Section 2, we present the framework for redescription set construction in Section 3. First, we shortly describe the CLUS-RM algorithm, then we introduce the conjunctive refinement procedure and explain the generalized redescription set construction process. Next, we introduce the variability index: which supports a refined treatment of redescription accuracy in presence of missing values. We describe the datasets and an application involving redescription sets produced by the framework in Section 4 and perform theoretical and empirical evaluation of the framework's performance in Section 5. Empirical evaluation includes quality analysis of representative sets and comparison to the set containing all discovered redescriptions, evaluation of the conjunctive refinement procedure, and quality comparison of redescriptions produced by our framework to those produced by several state of the art redescription mining algorithms, on three datasets with different properties. We conclude the paper in Section 6.

2. Notation and definitions

The input dataset $D = (V_1, V_2, E, W_1, W_2)$ is a quintuple of the two attribute (variable) sets (V_1, V_2), an element (instance) set E, and the two views corresponding to these attribute sets. Views (W_1 and W_2) are $|E| \times |V_d|$ data matrices such that $W_{d_{i,j}} = c_k$ if an element e_i has a value c_k for attribute $v_j \in V_d$.

A query q is a logical formula F that can contain the conjunction, disjunction and negation logical operators. These operators describe logical relations between different attributes, from attribute sets V_1 and V_2, that constitute a query. The set of all valid queries Q is called a query language. The set of elements described by a query q, denoted $supp(q)$, is called its support. A redescription $R = (q_1, q_2)$ is defined as a pair of queries, where q_1 and q_2 contain variables from V_1 and V_2 respectively. The support of a redescription is the set of elements supported by both queries that constitute this redescription $supp(R) = supp(q_1) \cap supp(q_2)$. We use $attr(R)$ to denote the multi-set of all occurrences of attributes in the queries of a redescription R. The corresponding set of attributes is denoted $attrs(R)$. The set containing all produced redescriptions is denoted $\mathcal{R}$. User-defined constraints $\mathcal{C}$ are typically limits on various redescription quality measures.

Given a dataset D, a query language Q over a set of attributes V, and a set of constraints $\mathcal{C}$, the task of redescription mining (Galbrun, 2013) is to find all redescriptions satisfying constraints in $\mathcal{C}$.

2.1. Individual redescription quality measures

The accuracy of a redescription $R = (q_1, q_2)$ is measured with the Jaccard similarity coefficient (Jaccard index).

$$J(R) = \frac{|supp(q_1) \cap supp(q_2))|}{|supp(q_1) \cup supp(q_2)|}$$

The problem with this measure is that redescriptions describing large subsets of instances often have a large intersection which results in high value of Jaccard index. As a result, the obtained knowledge is quite general and often not very useful to the domain expert. It is thus preferred to have redescriptions that reveal more specific knowledge about the studied problem and are harder to obtain by random sampling from the underlying data distribution.

This is why we compute the statistical significance (p-value) of each obtained redescription. We denote the marginal probability of a query q_1 and q_2 with $p_1 = \frac{supp(q_1)}{|E|}$ and $p_2 = \frac{supp(q_2)}{|E|}$, respectively and the set of elements described by both as $o = supp(q_1) \cap supp(q_2)$. The corresponding p-value (Galbrun, 2013) is defined as

$$pV(R) = \sum_{n=|o|}^{|E|} \binom{|E|}{n} (p_1 \cdot p_2)^n \cdot (1 - p_1 \cdot p_2)^{|E|-n}$$

The p-value represents a probability that a subset of elements of observed size or larger is obtained by joining two random queries with marginal probabilities equal to the fractions of covered elements. It is an optimistic criterion, since the assumption that all elements can be sampled with equal probability need not hold for all datasets.

Since it is important to provide understandable and short descriptions, it is interesting to measure the number of attributes occurring in redescription queries $attr(R)$.

Below, we provide an example of a redescription, together with its associated quality measures obtained on the Bio dataset (Mitchell-Jones, 1999; Hijmans et al., 2005; Galbrun, 2013):
Redescription $R'_{ex} = (q'_1, q'_2)$ with its queries defined as:
q'_1 : $(-1.8 \leq \tilde{t}_7 \leq 4.4 \wedge 12.1 \leq \tilde{p}_6 \leq 21.2) \vee$
$(-1.6 \leq \tilde{t}_6 \leq 1.5 \wedge 21.6 \leq \tilde{p}_6 \leq 30.1)$
q'_2 : Polarbear
describes 34 locations which are inhabited by the polar bear. The q'_1 query describes the average temperature ($\tilde{t}$) and the average precipitation ($\tilde{p}$) conditions of these locations in June and July. The redescription has a Jaccard index value of 0.895 and a p-value smaller than $2 \cdot 10^{-16}$. The multi-set $attr(R'_{ex}) = \{\tilde{t}_6, \tilde{t}_7, \tilde{p}_6, \tilde{p}_6, \text{Polarbear}\}$ and its corresponding set $attrs(R'_{ex}) = \{\tilde{t}_6, \tilde{t}_7, \tilde{p}_6, \text{Polarbear}\}$. The query size of R'_{ex}, denoted $|attr(R'_{ex})|$, equals 5.

2.2. Redescription quality measures based on redescription set properties

We use two redescription quality measures based on properties of redescriptions contained in a corresponding redescription set.

The measure providing information about the redundancy of elements contained in the redescription support is called the average redescription element Jaccard index and is defined as:

$$AEJ(R_i) = \frac{1}{|\mathcal{R}| - 1} \cdot \sum_{j=1}^{|\mathcal{R}|} J(supp(R_i), supp(R_j)),\ i \neq j$$

Analogously, the measure providing information about the redundancy of attributes contained in redescription queries, called the average redescription attribute Jaccard index, is defined as:

$$AAJ(R_i) = \frac{1}{|\mathcal{R}| - 1} \cdot \sum_{j=1}^{|\mathcal{R}|} J(attrs(R_i), attrs(R_j)),\ i \neq j$$

We illustrate the average attribute Jaccard index on the redescription example from the previous subsection. If we assume that our redescription set contains only two redescriptions $\mathcal{R} = \{R_{ex}, R'_{ex}\}$ where R_{ex} equals:

q_1 : $(-1.8 \leq \tilde{t}_7 \leq 4.4 \wedge 12.1 \leq \tilde{p}_6 \leq 21.2)$
q_2 : Polarbear

The corresponding average attribute Jaccard index of the redescription R_{ex} equals $\frac{3}{4} = 0.75$ showing a high level of redundancy in the used attributes between redescription R_{ex} and the only other redescription available in the set R'_{ex}. On the other hand, in the redescription set $\mathcal{R} = \{R'_{ex}, R''_{ex}\}$, where R''_{ex} contains queries:
q''_1 : $(7.2 \leq t^+_9 \leq 17.2 \wedge 13.5 \leq t^+_7 \leq 22.7)$
q''_2 : MountainHare

the average attribute Jaccard index of the redescription R'_{ex} equals $\frac{0}{7} = 0$ showing no redundancy in the used attributes.

3. Redescription mining framework

In this section, we present a redescription mining framework. It first creates a large set of redescriptions and then uses it to create one or more smaller sets that are presented to the user. This is done by taking into account the relative user preferences regarding importance of different redescription quality criteria.

3.1. The CLUS-RM algorihtm

The framework generates redescriptions with the CLUS-RM algorithm Mihelčić et al. (2015b), presented in Algorithm 1. It uses multi-target Predictive Clustering Trees (PCT) (Kocev et al., 2013) to construct conjunctive queries which are used as building blocks of redescriptions. Queries containing disjunctions and negations are obtained by combining and transforming queries containing only conjunction operator.

Algorithm 1 The CLUS-RM algorithm

```
Require: First view data ($W_1$), Second view data ($W_2$),
    Constraints $\mathcal{C}$
Ensure: A set of redescriptions $\mathcal{R}$
1: procedure CLUS-RM
2:     $[P_{W1init}, P_{W2init}] \leftarrow$ createInitialPCTs($W_1$, $W_2$)
3:     $[r_{W1}, r_{W2}] \leftarrow$ extrRulesFromPCT($P_{W1init}, P_{W2init}$)
4:     while RunInd<maxIter do
5:         $[D_{W1}, D_{W2}] \leftarrow$ constructTargets($r_{W1}$,$r_{W2}$)
6:         $[P_{W1}, P_{W2}] \leftarrow$ createPCTs($D_{W1}, D_{W2}$)
7:         extractRulesFromPCT($P_{W1}, P_{W2}, r_{W1}, r_{W2}$)
8:         $\mathcal{R} \leftarrow \mathcal{R} \cup$ createRedescriptons($r_{W_1}, r_{W_2}$, $\mathcal{C}$)
9:     return $\mathcal{R}$
```

The algorithm is able to produce a large number of highly accurate redescriptions from which many contain only conjunction operator in the queries. This is in part the consequence of using PCTs in multi-target setting, which is known to outperform single class classification or regression trees due to the property of inductive transfer (Piccart, 2012). This distinguishes the CLUS-RM redescription mining algorithm from other state of the art solutions that in general create a smaller number of redescriptions with majority of redescription queries containing the disjunction operator.

3.1.1. Rule construction and redescription creation

The initial task in the algorithm is to create one PCT per view of the original data, constructed for performing unsupervised tasks, to obtain different subsets of instances (referred to as initial clusters) and the corresponding queries that describe them. To create initial clusters (line 2 in Algorithm 1), the algorithm transforms an unsupervised problem to a supervised problem by constructing an artificial instance for each original instance in the dataset. These instances are obtained by shuffling attribute values among original instances thus braking any existing correlations between the attributes. Each artificial instance is assigned a target label 0.0 while each original instance is assigned a target label 1.0. One such dataset is created for each view considered in the redescription mining process. A PCT is constructed on each dataset, with the goal of distinguishing between the original and the artificial instances, and transformed to a set of rules. This transformation is achieved by traversing the tree, joining all attributes used in splits into a rule and computing its support. Each node in a tree forms one query containing the conjunction and possibly negation operators (line 3 and 7 in Algorithm 1).

After the initial queries are created, the algorithm connects different views by assigning target labels to instances based on their coverage by queries constructed from the opposing view (line 5 in Algorithm 1). To construct queries containing attributes from W_2, each instance is assigned a target label 1.0 if it is described by a query containing the attributes from W_1, otherwise it is assigned a value 0.0. The process is iteratively repeated a predefined number of steps (line 4 in Algorithm 1).

Redescriptions are created as a Cartesian product of a set of queries formed on W_1 and a set of queries formed on W_2 (line 8 in Algorithm 1). All redescriptions that satisfy user defined constraints ($\mathcal{C}$): the minimal Jaccard index, the maximal p-value, the minimal and the maximal support are added to the redescription set. The algorithm can produce redescriptions containing conjunction, negation and disjunction operators.

The initialization, rule construction and various types of redescription creation are thoroughly described in (Mihelčić et al., 2015b).

3.1.2. Conjunctive refinement

In this subsection, we present an algorithmic improvement to the redescription mining process presented in Algorithm 1. The aim of this method is to improve the overall accuracy of redescriptions in the redescription set by combining newly created redescriptions with redescriptions already present in redescription set $\mathcal{R}$.

Combining existing redescription queries with an attribute by using conjunction operator has been used in greedy based redescription mining algorithms (Gallo et al., 2008; Galbrun & Miettinen, 2012b) to construct redescriptions. The idea is to expand each redescription query in turn by using a selected attribute and the selected logical operator. Such procedure, if used with the conjunction operator, leads to increase of Jaccard index but also mostly reduces the support size of a redescription. Zaki & Ramakrishnan (2005) combine closed descriptor sets by using conjunction operator to construct a closed lattice of descriptor sets which are used to construct redescriptions. They conclude that combining descriptor set D_1 and D_2 describing element sets G_1 and G_2 respectively, such that $G_1 \subseteq G_2$, can be done by constructing a descriptor set $D_1 \cup D_2$. They conclude that the newly created descriptor set, describes the same set of elements G_1 as the set D_1. This procedure works only with attributes containing Boolean values and does not use the notion of views.

Instead of extending redescription queries with attributes connected using conjunction operator (which is usually constrained by the number of expansions), the conjunctive refinement procedure compares support of each redescription $R = (q_1, q_2)$ in the redescription set with the selected redescription $R_{ref} = (q'_1, q'_2)$. It merges the queries of these two redescriptions with the $\{\wedge\}$ operator to obtain a new redescription $R_{new} = (q_1 \wedge q'_1, q_2 \wedge q'_2)$ if and only if $supp(R) \subseteq supp(R_{ref})$. We extend and prove the property described in Zaki & Ramakrishnan (2005) in a more general setting, combining redescriptions with arbitrary type of attributes and a finite amount of different views. We demonstrate how to use it efficiently with numerical attributes and show that this procedure does not decrease the accuracy of a redescription. In fact, if $\exists e \in E,\ e \in supp(q_1)\ \vee\ \exists e' \in E,\ e' \in supp(q_2)$ such that $e \notin supp(q'_1) \vee\ e' \notin supp(q'_2)$, than $J(R_{new}) > J(R)$.

If the attributes contain numerical values, we can transform the redescription R_{ref}, given an arbitrary redescription $R \in \mathcal{R}$ such that $supp(R) \subseteq supp(R_{ref})$, to redescription $R'_{ref} = (q''_1, q''_2)$ such that R'_{ref} has tighter numerical bounds on all attributes contained in the queries, $supp(R) \subseteq supp(R'_{ref})$ and that $J(supp(R), supp(R'_{ref})) \geq J(supp(R), supp(R_{ref}))$. By doing this, we increase the probability of finding the element e or e' as described above, which leads to improving the accuracy of redescription R_{new}. The construction procedure of such redescription is explained in Section S1.1 (Online Resource 1). The redescription R'_{ref} is used as a refinement redescription when numerical attributes are present in the data.

We can now state and prove the following lemma:

Lemma 3.1. *For every redescription $R \in \mathcal{R}$, for every redescription $R_{ref} = (q'_1, q'_2)$, where $q'_1 = q_{a_1} \wedge q_{a_2} \wedge \ldots \wedge q_{a_n}$, $a_i \in attrs(R_{ref})$, $\forall i \in \{1, \ldots, n\}$ and $n \in \mathbb{N}$, $q'_2 = q_{b_1} \wedge q_{b_2} \wedge \ldots \wedge q_{b_m}$, $b_j \in attrs(R_{ref})$, $\forall j \in \{1, \ldots, m\}$ and $m \in \mathbb{N}$. If $supp(R) \subseteq supp(R_{ref})$ then for a redescription $R_{new} = (q_1 \wedge q'_1, q_2 \wedge q'_2)$ it holds that $J(R_{new}) \geq J(R)$ and $supp(R_{new}) = supp(R)$.*

The proof of Lemma 3.1 for redescription mining problems containing two views can be seen in Section S1.1 (Online Resource 1). General formulation with n arbitrary views is proven by mathematical induction. It is easily seen from the proof that if $\exists e \in E,\ e \in supp(q_1) \ \vee\ \exists e' \in E,\ e' \in supp(q_2)$ such that $e \notin supp(q'_1) \vee\ e' \notin supp(q'_2)$ then $supp(q_1 \wedge q'_1) \cup supp(q_2 \wedge q'_2) \subset supp(q_1) \cup supp(q_2)$ thus ultimately $J(R_{new}) > J(R)$.
The conjunctive refinement is demonstrated in Figure 2.

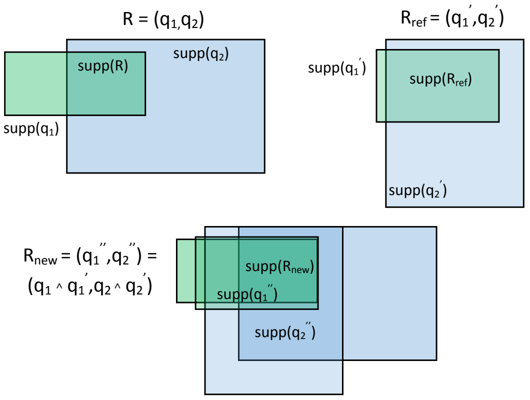


Figure 2: Demonstration of the effects of the conjunctive refinement on a support of the improved redescription and corresponding redescription queries. For the supports represented on the figure it holds: $supp(R) \subset supp(R_{ref})$. As a consequence: $supp(R) = supp(R_{new})$, $J(R_{new}) > J(R)$.

Line 8 from Algorithm 1 is replaced with the procedure $\mathcal{R} \leftarrow$*createAndRefineRedescriptions*$(rw_1, rw_2, \mathcal{R}, \mathcal{C})$ which is presented in Algorithm 2.

Algorithm 2 The redescription set refinement procedure

```
Require: Rules created on $W_1$ ($rw_1$), Rules created on $W_2$ ($rw_2$), Redescription set $\mathcal{R}$, Constraints $\mathcal{C}$
Ensure: A set of redescriptions $\mathcal{R}$
procedure ConstructAndRefine
    for $R_{new} \in rw_1 \times rw_2$ do
        if $R_{new}.J \geq \mathcal{C}.minJref$ then
            for $R \in \mathcal{R}$ do
                $R.Refine(R_{new})$
                $R_{new}.Refine(R)$
            if $R_{new}.J \geq \mathcal{C}.minJ$ then
                $\mathcal{R} \leftarrow \mathcal{R} \cup R_{new}$
    return $\mathcal{R}$
```

The procedure described in Algorithm 2 and demonstrated in Figure S1 applies conjunctive refinement by using redescriptions that satisfy the user defined constraints $\mathcal{C}$ and redescriptions that satisfy looser constraints on the Jaccard index ($R.J \geq C.minRefJ$, $C.minRefJ \leq C.minJ$). These constraints determine the amount and variability of redescriptions used to improve the redescription set.

The refinement procedure, in combination with redescription query minimization explained in Mihelčić et al. (2015b), provides grounds for mining more accurate yet compact redescriptions.

3.2. Generalized redescription set construction

The redescription set obtained by Algorithm 1 contains redescriptions satisfying hard constraints described in the previous subsections. It is often very large and hard to explore. For this reason, we extract one or more smaller sets of redescriptions that satisfy additional preferential properties on objective redescription evaluation measures, set up by the user, and present them for exploration. This process is demonstrated in Figure 3.

Producing summaries and compressed rule set representations is important in many fields of knowledge discovery. In the field of frequent itemset mining such dense representations include closed itemsets (Pasquier et al., 1999) and free sets (Boulicaut & Bykowski, 2000). The approaches using set pattern mining construct a set by enforcing constraints on different pattern properties, such as support, overlap or coverage (Guns et al., 2011). Methods developed in information theory consider sets that provide the best compression of a larger set of patterns. These techniques use properties like the Information Bottleneck (Tishby et al., 1999) or the Minimum description length (Grünwald, 2007). The work on statistical selection of association rules developed by Bouker et al. (2012) presented techniques to eliminate irrelevant rules based on dominance, which is computed on several possibly conflicting criteria. If some rule is not strictly dominated by any other rule already in the set, the minimal similarity

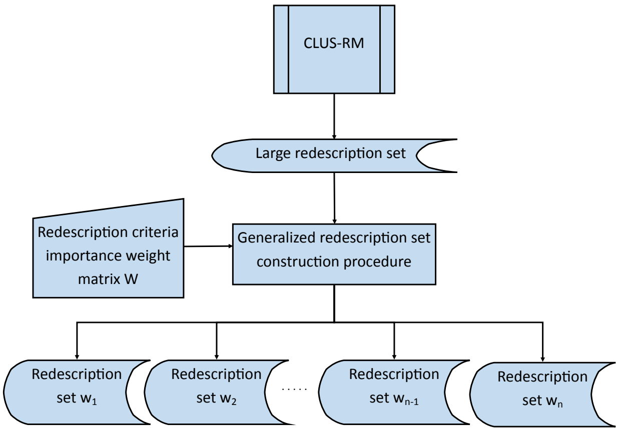

CLUS-RM
Large redescription set
Redescription criteria importance weight matrix W
Generalized redescription set construction procedure
Redescription set w$_1$
Redescription set w$_2$
.....
Redescription set w$_{n-1}$
Redescription set w$_n$

Figure 3: Flowchart representing the redescription set construction process.

with some representative rule is used to determine if it should be added to the set.

Redescriptions are highly overlapping with respect to described instances and attributes used in the queries. It is often very hard to find fully dominated redescriptions, and the number of dominated redescriptions that can be safely discarded is relatively small compared to a set of all created redescriptions. Our approach, to create a set of user defined (small) size, does not use a representative rule to compute the similarity. Instead, it adds redescriptions to the final redescription set by using the scalarization technique (Caramia & Dell'Olmo, 2008) developed in multi-objective optimization to find the optimal solution when faced with many conflicting criteria. If the corresponding optimization function is minimized, given positive weights, the solution is a strict pareto optimum, otherwise it is a weak pareto optimum (Caramia & Dell'Olmo, 2008) of a multi objective optimization problem. Similar aggregation technique is used in multi attribute utility theory - MAUT (Winterfeldt & Fischer, 1975) to rank the alternatives in decision making problems.

Each redescription is evaluated with a set of criteria known from the literature or defined by the user. The final quality score is obtained by aggregating these criteria with user-defined importance weights to produce a final numerical score. Based on this score, the method selects one non-dominated redescription, based on utilised quality criteria, at each step of redescription set construction.

The procedure generalizes the current redescription set construction approaches in two ways: 1) it allows defining importance weights to different redescription quality criteria and adding new ones to enable constructing redescription sets with different properties which provides different insight into the data, 2) it allows creating multiple redescription sets by using different weight vectors, support levels, Jaccard index thresholds or redescription set sizes. Thus, it in many cases eliminates the need to make multiple runs of a redescription mining algorithm.

One extremely useful property of the procedure is that it can be used by any existing redescription mining algorithm, or a combination thereof. In general, larger number of diverse, high quality redescriptions allows higher quality reduced sets construction.

Are there any elements in the data that share many common properties? Can we find a subset of elements that allows multiple different redescriptions? Can we find very diverse but accurate redescriptions? What is the effect of reducing redescription query size to the overall accuracy on the observed data? What are the effects of missing values to the redescription accuracy? What is our confidence that these redescriptions will remain accurate if missing values are added to our set? This is only a subset of questions that can be addressed by observing redescription sets produced by the proposed procedure. The goal is not to make redescription mining subjective in the sense of interestingness (Tuzhilin, 1995) or unexpectedness (Padmanabhan & Tuzhilin, 1998), but to enable exploration of mined patterns in a more versatile manner.

The input to the procedure is a set of redescriptions produced by Algorithm 1 and an importance weight matrix defined by the user. The rows of the importance weight matrix define the users' importance for various redescription quality criteria. The procedure creates one output redescription set for each row in the importance weight matrix (line 3 in Algorithm 3). The procedure works in two parts: first it computes element and attribute occurrence in redescriptions from the original redescription set (line 2 in Algorithm 3). This information is used to find the redescription that satisfies the user defined criteria and describes elements by using attributes that are found in a small number of redescriptions from the redescription set. When found (line 4 in Algorithm 3), it is placed in the redescription set being constructed (line 5 in Algorithm 3). Next, the procedure iteratively adds non-dominated redescriptions (lines 7-9 in Algorithm 3) until the maximum allowed number of redescriptions is placed in the newly constructed set (line 6 in Algorithm 3).

Algorithm 3 Generalized redescription set construction

```
Require: Redescription set $\mathcal{R}$, Importance weight matrix $\mathcal{W}$, Size of reduced set $n$
Ensure: A set of reduced redescription sets $\mathcal{R}_{red}$
 1: procedure ReduceSet
 2:     $[E_{ocur}, A_{ocur}] \leftarrow$ computeCoocurence($\mathcal{R}$)
 3:     for $w_i \in \mathcal{W}$ do
 4:         $R_{first} \leftarrow$ findSpecificRed($\mathcal{R}, E_{cooc}, A_{cooc}, w_i$)
 5:         $\mathcal{R}_{w_i} \leftarrow \mathcal{R}_{w_i} \cup R_{first}$
 6:         while $|R_{w_i}| < n$ do
 7:             $R_{best} \leftarrow$ findBest($\mathcal{R}, \mathcal{R}_{w_i}, w_i$)
 8:             $\mathcal{R}_{w_i} \leftarrow \mathcal{R}_{w_i} \cup R_{best}$
 9:         $\mathcal{R}_{red} \leftarrow \mathcal{R}_{red} \cup \{\mathcal{R}_{w_i}\}$
10:     return $\mathcal{R}_{red}$
```

In the current implementation, we use 6 redescription quality criteria, however more can be added. Five of these criteria are general redescription quality criteria, the last one is used when the underlying data contains missing values and will be described in the following section.

The procedure *findSpecificRed* uses the information about the redescription Jaccard index, p-value, query size and the occurrence of elements described by the redescription and attributes found in redescriptions queries in redescriptions from the redescription set. The p-value quality score of a redescription R is computed as:

$$score_{pval}(R) = \begin{cases} \frac{log_{10}(pV(R))}{17} + 1 & , pV(R) \geq 10^{-17} \\ 0 & , pV(R) < 10^{-17} \end{cases}$$

The logarithm is applied to linearise the p-values and the normalization 17 is used because 10^{-17} is the smallest possible p-value that we can compute.

The element occurrence score of a redescription is computed as: $score_{ocurEl}(R) = \frac{\sum_{e_k \in supp(R)} E_{ocur}[k]}{\sum_{j=1}^{|E|} E_{ocur}[j]}$. The attribute occurrence score is computed in the same way as: $score_{ocurAt}(R) = \frac{\sum_{a_k \in attrs(R)} A_{ocur}[k]}{\sum_{j=1}^{|V_1|+|V_2|} A_{ocur}[j]}$. We also compute the score measuring query size in redescriptions:

$$score_{size} = \begin{cases} \frac{|attr(R)|}{k} & , |attr(R)| < k \\ 1 & , k \leq |attr(R)| \end{cases}$$

The user-defined constant k denotes redescription complexity normalization factor. In this work we use $k = 20$, because redescriptions containing more than 20 variables in the queries are highly complex and hard to understand.

The first redescription is chosen by computing: $R_{first} = argmin_R\ (w_0 \cdot (1.0 - J(R)) + w_1 \cdot score_{pval}(R) + w_2 \cdot score_{ocurEl}(R) + w_3 \cdot score_{ocurAt}(R) + w_4 \cdot score_{size}(R))$. Each following redescription is evaluated with a score function that computes redescription similarity to each redescription contained in the redescription set. The similarity is based on described elements and attributes used in redescription queries. This score thus allows controlling the level of redundancy in the redescription set. For a redescription $R_i \in \mathcal{R} \backslash \mathcal{R}_{red}$ we compute: $score_{elemSim}(R_i) = max_j\ J(supp(R_i), supp(R_j)),\ j = 1, \ldots, |\mathcal{R}_{red}|$ and $score_{attrSim}(R) = max_j\ J(attrs(R_i), attrs(R_j)),\ j = 1, \ldots, |\mathcal{R}_{red}|$.

Several different approaches to reducing redundancy among redescriptions have been used before, however no exact measure was used to select redescriptions or to assess the overall level of redundancy in the redescription set. Zaki & Ramakrishnan (2005) developed an approach for non-redundant redescription generation based on a lattice of closed descriptor sets, Ramakrishnan et al. (2004) used the parameter defining the number of times one class or descriptor is allowed to participate in a redescription. This is used to make a trade-off between exploration and redundancy. Parida & Ramakrishnan (2005) computed non-redundant representations of sets of redescriptions containing some selected descriptor (set of Boolean attributes). Galbrun & Miettinen (2012b) defined a minimal contribution parameter each literal must satisfy to be incorporated in a redescription query. This enforces control over redundancy on the redescription level. Redundancy between different redescriptions is tackled in the Siren tool Galbrun & Miettinen (2012c) as a post processing (filtering) step. Mihelčić et al. (2015b) use weighting of attributes occurring in redescription queries and element occurrence in redescription supports based on work in subgroup discovery (Gamberger & Lavrac, 2002; Lavrač et al., 2004).

We combine the redescription p-value score with its support to first add highly accurate, significant redescriptions with smaller support, and then incrementally add accurate redescriptions with larger support size. Candidate redescriptions are found by computing: $R_{best} = argmin_R\ (w_0 \cdot (1.0 - J(R)) + w_1 \cdot (\frac{k}{n} \cdot score_{pval}(R) + (1 - \frac{k}{n}) \cdot \frac{supp(R)}{|E|}) + w_2 \cdot score_{elemSim}(R) + w_3 \cdot score_{attrSim}(R) + w_4 \cdot score_{size}(R))$, where k denotes the number of redescriptions contained in the set under construction at this step.

3.3. Missing values

There are more possible ways of computing the redescription Jaccard index when the data contains missing values. The approach that assumes that all elements from redescription support containing missing values are distributed in a way to increase the redescription Jaccard index is called optimistic (J_{opt}). Similarly, the approach that assumes that all elements from redescription support containing missing values are distributed in a way to decrease the redescription Jaccard index is called pessimistic (J_{pess}). The rejective Jaccard index evaluates redescriptions only by observing elements that do not contain missing values for attributes contained in redescription queries. These measures are discussed in (Galbrun & Miettinen, 2012b). The Query non-missing Jaccard index (J_{qnm}), introduced in (Mihelčić et al., 2015b), is an approach that gives a more conservative estimate than the optimistic Jaccard index but more optimistic estimate than the pessimistic Jaccard index. The main evaluation criteria for this index is that a query (containing only the conjunction operator) can not describe an element that contains missing values for attributes in that query. This index is by its value closer to the optimistic than the pessimistic Jaccard index. However, as opposed to the optimistic approach, redescriptions evaluated by this index contain in their support only elements that have defined values for all attributes in redescription queries and that satisfy query constraints. The index does not penalize the elements containing missing values for attributes in both queries which are penalized in the pessimistic Jaccard index.

In this paper, we introduce a natural extension to the presented measures: the redescription variability index. This index measures the maximum possible variability in redescription accuracy due to missing values. This allows finding redescriptions that have only slight variation in accuracy regardless the actual value of the missing values. It also allows reducing very strict constraints imposed by the pessimistic Jaccard index that might lead to the elimination of some useful redescriptions.

The redescription variability index is defined as: $\mathit{variability}(R) = J_{opt}(R) - J_{pes}(R)$.
Formal definitions of pessimistic and optimistic Jaccard index can be seen in Section S1.2 (Online resource 1).

The scores used to find the first and the best redescription in generalized redescription set construction (Section 3.2) are extended to include the *variability* score.
Our framework optimizes query non-missing Jaccard but reports all Jaccard index measures when mining redescriptions on the data containing missing values. In principle with the generalized redescription set construction, we can return reduced sets containing accurate redescriptions found with respect to each Jaccard index. Also, with the use of variability index, the framework allows finding redescriptions with accuracy affected to a very small degree by the missing values which is not possible by other redescription mining algorithms in the literature. The only approach working with missing values ReReMi requires preforming multiple runs of the algorithm to make any comparisons between redescriptions mined by using different version of Jaccard index.

4. Data description and applications

We describe three datasets used to evaluate CRM-GRS and demonstrate its application on a Country dataset.

4.1. Data description

The evaluation and comparisons are performed on three datasets with different characteristics: the Country dataset (UNCTAD, 2014; WorldBank, 2014; Gamberger et al., 2014), the Bio dataset (Mitchell-Jones, 1999; Hijmans et al., 2005; Galbrun, 2013) and the DBLP dataset (DBLP, 2010; Galbrun, 2013). Detailed description of each dataset can be seen in Section S2 (Online resource 1).

Table 1: Description of datasets used to perform experiments

Dataset	W_1 **attributes**	W_2 **attributes**
Country $\lvert E\rvert = 199$ countries	Numerical (49) World Bank Year: 2012 Country info	Numerical (312) UNCTAD Year: 2012 Trade Info
Bio $\lvert E\rvert = 2575$ geographical locations	Numerical (48) Climate conditions	Boolean (194) mammal species
DBLP $\lvert E\rvert = 6455$ authors	Boolean (304) author-conference bi-partite graph	Boolean (6455) co-authorship network

Descriptions of all attributes used in the datasets are provided in the document (Online Resource 2).

4.2. Application on the Country dataset

The aim of this study is to discover regularities and interesting descriptions of world countries with respect to their trading properties and general country information (such as various demographic, banking and health related descriptors). We will focus on redescriptions describing four European countries: Germany, Czech Republic, Austria and Italy, discovered as a relevant cluster in a study performed by Gamberger et al. (2014). This study investigated country and trade properties of EU countries with potential implications to a free trade agreement with China. This or similar use-case may be a potential topic of investigation for economic experts but the results of such analysis could also be of interest to the policymakers and people involved in export or import business.

First step in the exploration process involves specifying various constraints on produced redescriptions. Determining parameters such as minimal Jaccard index or minimal support usually requires extensive experimentation. These experiments can be performed with CRM-GRS with only one run of redescription mining algorithm by using minimal Jaccard index of 0.1, minimal support of 5 countries (if smaller subsets are not desired) and p-value of 0.01. Parameters specifying reduced set construction can now be tuned to explore different redescription set sizes, minimal Jaccard thresholds or minimal and maximal support intervals. Results of such meta analysis (presented in Section S2.2.2 (Online resource 1)) show little influence of setting minimal Jaccard threshold on this dataset, however right choice of minimal support is important. Redescription sets using minimal support threshold of 5 countries show superior properties and may contain useful knowledge.

We present three different redescriptions describing specified countries and revealing their similarity to several other countries (demonstrated in Figure 4).

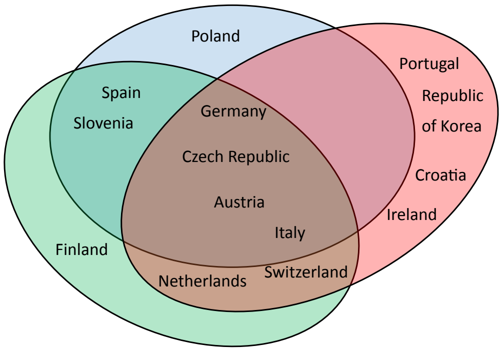


Figure 4: Similarities between different, mostly European, countries.

Redescriptions R_{blue}, R_{green} and R_{red} are defined as:
$qb_1: \ 13.2 \leq \ \mathrm{POP}_{14} \leq 15.2 \ \wedge \ 3.1 \leq MORT \leq 5.0 \ \wedge 0.0 \leq \ \mathrm{POP\_GROWTH} \ \leq 0.5$
$qb_2: \ 13.2 \leq \mathrm{E/I\_MiScManArt} \leq 15.2 \ \wedge \ 28.0 \leq \mathrm{E\_MedSTehInMan} \leq 40.0.$
$(J_{qnm}(R_{blue}) = J_{opt}(R_{blue}) = 1.0, \ J_{pess}(R_{blue}) = 0.88, \ pV(R_{blue}) = 2.3 \cdot 10^{-10}, \ |supp(R_{blue})| = 7)$
$qg_1: \ 16.2 \leq \ \mathrm{POP}_{64} \leq 21.1 \ \wedge \ 2.9 \leq MORT \leq 4.5 \ \wedge 16.2 \leq \ \mathrm{RUR\_POP} \ \leq 50.1 \ \wedge \ 0.2 \leq \mathrm{W\_REM} \leq 1.4$
$qg_2: \ 0.8 \leq \mathrm{E/I\_ElMachApp} \leq 1.8 \ \wedge \ 93.0 \leq \mathrm{E\_AlocProd} \leq 99.0 \ \wedge \ 1.1 \leq \mathrm{E/I\_SpecMach} \leq 4.3.$
$(J_{qnm}(R_{green}) = J_{opt}(R_{green}) = J_{pess}(R_{green}) = 1.0, \ pV(R_{green}) = 1.9 \cdot 10^{-11}, \ |supp(R_{blue})| = 9)$
$qr_1: \ 3.6 \leq \ \mathrm{MORT} \ \leq 4.7 \wedge 22.9 \leq \mathrm{CRED\_COV} \leq 100.0 \ \wedge 77.3 \leq \ \mathrm{M2} \leq 238.9$
$qr_2: \ 0.1 \leq \mathrm{E/I\_Cereals} \leq 1.7 \ \wedge \ 1.2 \leq \mathrm{E/I\_BevTob} \leq 3.1 \ \wedge \ 0.7 \leq \mathrm{E/I\_SpecMach} \leq 4.3.$
$(J_{qnm}(R_{red}) = J_{opt}(R_{red}) = 1.0, \ J_{pess}(R_{red}) = 0.45, \ pV(R_{red}) = 6.3 \cdot 10^{-12}, \ |supp(R_{red})| = 10)$

Table 2: Description of attributes from R_{blue}, R_{green} and R_{red}

Code	Description
POP_{14}	% of population aged [0,14]
POP_{64}	% of population aged 65+
$MORT$	Mortality under 5 years per 1000
POP_GROWTH	% of population growth
RUR_POP	% of population living in rural area
W_REM	% of GDP spent on worker's remittances and compensation
$CRED\_COV$	% of adults listed by private credit bureau
M_2	% of GDP as (quasi) money
$E, \ I, \ E/I$	export, import, export to import ratio
$MiScManArt$	Miscellaneous manufactured articles
$MedSTehInMan$	Medium - skill, technology - intensive manufactures
$ElMachApp$	Electrical machinery, apparatus and appliances
$AlocProd$	All allocated products
$SpecMach$	Specialised machinery
$Cereals$	Cereals and cereal preparations
$BevTob$	Beverages and tobacco

Presented redescriptions (attribute descriptions available in Table 2) confirm several findings reported in (Gamberger et al., 2014). Mainly, high export of medium - skill and technology - intensive manufactures, export of beverages and tobacco, low percentage of young population. Additionally, these redescriptions reveal high percentage of elderly population (age 65 and above), lower (compared to world average of 47.4) but still present mortality rate of children under 5 years of age (per 1000 living) and small to medium percentage of rural population. The credit coverage (percentage of adults registered for having unpaid depths, repayment history etc.) varies between countries but is no less than 20% adult population. The money and quasi money (M2 - sum of currency outside banks etc.) is between substantial 77.3% and very large 239% of total country's GDP. For additional examples see Section S2.2.3, Figure S11 (Online resource 1).

Output of CRM-GRS can be further analysed with visualization and exploration tools such as the Siren (Galbrun & Miettinen, 2012c) (available at `http://siren.gforge.inria.fr/main/`) or the InterSet (Mihelčić & Šmuc, 2016) (available at `http://zel.irb.hr/interset/`). In particular, the InterSet tool allows exploration of different groups of related redescriptions, discovery of interesting associations, multi-criteria filtering and redescription analysis on the individual level.

5. Evaluation and comparison

In this section we present the results of different evaluations. First, we perform a theoretical comparison of our approach with other state of the art solutions which, includes description of advantages and drawbacks of our method. Next, we apply the generalized redescription set construction procedure to these datasets starting from redescriptions created by the CLUS-RM algorithm. We evaluate the conjunctive refinement procedure and perform a thorough comparison of our reduced sets with the redescription sets obtained by several state of the art redescription mining algorithms. The comparisons use measures on individual redescriptions (Section 2.1) as well as measures on redescription sets (Section 2.2). We also use the normalized query size defined in Section 3.2.

The execution time analysis, showing significant time reduction when using generalized redescription set construction instead of multiple CLUS-RM runs, is described in Section S2.4 (Online resource 1).

5.1. Theoretical algorithm comparison

We compare the average case time and space complexity of the CRM-GRS with state of the art approaches and present the strengths and weaknesses of our framework. The term $z = 2^d - 1$ in Table 3 denotes the number of nodes in the tree and is constrained by the tree depth d. $\mathcal{C}$ denotes the set of produced maximal closed frequent itemsets, l denotes the length of the longest itemset, $\mathcal{B}$ a set of produced biclusters, $L = \sum_{c\in\mathcal{B}} |c|$ and $\mathcal{R}$ denotes a set of produced redescriptions.

We can see from Table 3 that the CRM-GRS has slightly higher computational complexity than other tree - based approaches (which is based on time complexity of algorithm C4.5), caused by complexity of underlying redescription mining algorithm CLUS-RM. Optimizations proposed in (Mihelčić et al., 2015b) lower average time complexity of basic algorithm to $O(z \cdot (|V_1| + |V_2|) \cdot |E|^2$ and algorithm with refinement to $O(z \cdot (|V_1| + |V_2|) \cdot |E|^2 + z^2 \cdot |E|)$.

Table 3: Time and space complexity of redescription mining algorithms and the generalized redescription set construction procedure

Algorithm	Time comp.	Space comp.
CRM-GRS	$O(z \cdot (\lvert V_1\rvert + \lvert V_2\rvert) \cdot \lvert E\rvert^2 + z^2 \cdot \lvert E\rvert)$ (No refinement) $O(z \cdot (\lvert V_1\rvert + \lvert V_2\rvert) \cdot \lvert E\rvert^2 + z^3 \cdot \lvert E\rvert)$ (refinement)	$O(z)$
CARTWh.	$O(z \cdot (\lvert V_1\rvert + \lvert V_2\rvert) \cdot \lvert E\rvert^2)$	$O(z)$
Split trees	$O(z \cdot (\lvert V_1\rvert + \lvert V_2\rvert) \cdot \lvert E\rvert^2)$	$O(z)$
Layered trees	$O(z \cdot (\lvert V_1\rvert + \lvert V_2\rvert) \cdot \lvert E\rvert^2)$	$O(z)$
Greedy	$O(\lvert V_1\rvert \cdot \lvert V_2\rvert \cdot \lvert E\rvert)$	$O(1)$
MID	$O(\lvert \mathcal{C}\rvert \cdot \lvert E\rvert \cdot 2^l)$	$O(1)$
Closed Dset	$O(\lvert \mathcal{C}\rvert \cdot \lvert E\rvert \cdot 2^l)$	$O(\lvert \mathcal{C}\rvert)$
Relaxation Latt.	$max(O(\lvert \mathcal{B}\rvert \cdot log (\lvert E\rvert) + (\lvert V_1\rvert + \lvert V_2\rvert) \cdot \lvert E\rvert, O(L \cdot log(\lvert E\rvert) + (\lvert V_1\rvert + \lvert V_2\rvert) \cdot \lvert E\rvert))$	$O(\lvert \mathcal{B}\rvert)$
GRSC	$O(\lvert \mathcal{R}\rvert \cdot \lvert E\rvert)$	$O(\lvert \mathcal{R}\rvert)$

Worst-case complexity with the use of refinement is $O(z \cdot (|V_1| + |V_2|) \cdot |E|^2 + z^4 \cdot |E|)$. It is the result of a very optimistic estimate that produced redescriptions satisfying user constraints grow quadratically with the number of nodes in the tree (this is only the case if no constraints on redescriptions are enforced). In reality, it has at most linear growth. Furthermore, term $z^2 \cdot |E|$ is only dominating if $z > (|V_1| + |V_2|) \cdot |E|$. Since redescription queries become very hard to understand if they contain more than 10 attributes, even with 2 attributes in each of two views, this term is dominated when $|E| > 255$ instances.

Greedy approaches (Gallo et al., 2008; Galbrun & Miettinen, 2012b) are less affected by the increase in number of instances than the tree-based approaches, but are more sensitive to the increase in number of attributes.

Complexity of approaches based on closed and frequent itemset mining (Gallo et al., 2008; Zaki & Ramakrishnan, 2005) depends on the number of produced frequent or closed itemsets which in worst case equals $2^{|V_1|+|V_2|}$. Similarly, the complexity of approach proposed by Parida & Ramakrishnan (2005) depends on the number of created biclusters and their size.

One property of our generalized redescription set construction procedure (GRSC) is that it can be used to replace multiple runs of expensive redescription mining algorithms. Analysis from Table 3 and in S2.6 (Online resource 1) shows that it has substantially lower time complexity than all state of the art approaches except the MID and the Closed Dset. However, even for this approaches, it might be beneficial to use GRSC instead of multiple runs of these algorithms when $|\mathcal{C}| \cdot 2^l > |\mathcal{R}|$.

Since a trade-off between space and time complexity can be made for each of the analysed algorithms, we write the space complexity as a function of stored itemsets, rules, redescriptions or clusters. To reduce execution time, these structures can be stored in memory together with corresponding instances which increases space complexity to $O(C_{old} \cdot |E|)$ for all approaces.

One drawback of our method is increased memory consumption ($O(z^2)$ in the worst case). Since we memorize all distinct created redescriptions that satisfy user constraints, it is among more memory expensive approaches. Although, the estimate $O(z^2)$ is greatly exaggerated, and is in real applications at most $O(z)$, it is currently the only approach that memorizes and uses all created redescriptions to create diverse and accurate redescription sets for the end users. If memory limit is reached, we use the GRCS procedure (called in line 8 of Algorithm 1) to create reduced redescription sets of predefined properties. Only redescriptions from these sets are retained allowing further execution of the framework.

Greedy and the MID approaches are very memory efficient since they store only a small number of candidate redescriptions in memory. Other tree-based approaches store two decision trees at each iteration, Closed Dset (Zaki & Ramakrishnan, 2005) approach saves a closed lattice of descriptor sets and the relaxation lattice approach (Parida & Ramakrishnan, 2005) saves produced biclusters.

The main advantages of our approach are that it produces a large number of diverse, highly accurate redescriptions which enables our multi-objective optimization procedure to generate multiple, high quality redescription sets of differing properties that are presented to the end user.

5.2. Experimental procedure

In this section we explain all parameter settings used to perform evaluations and comparisons with various redescription mining algorithms.

For all algorithms, we used the maximal p-value threshold of 0.01 (the strictest significance threshold). The minimal Jaccard index was set to 0.2 for the DBLP dataset based on results presented in Galbrun (2013), Table 6.1, p. 46. The same is set to 0.6 for the Bio dataset based on results in Galbrun (2013) Table 7, p. 301. The threshold 0.5 for the Country dataset was experimentally determined. Minimal support was set to 10 elements for the DBLP, based on Galbrun (2013) p.48, and the same is used for the Bio dataset. Country dataset is significantly smaller thus we set this threshold to 5 elements. Impact of changing minimal Jaccard index and minimal support is data dependant. Increasing these thresholds causes a drop in diversity of produced redescriptions, resulting in high redundancy and in some cases inadequate number of produced redescriptions. However, it also increases minimal and average redescription Jaccard index and support size. Lowering these thresholds has the opposite effect, increasing diversity but potentially reducing overall redescription accuracy or support size. Increasing maximal p-value threshold allows more redescriptions (although less significant) to be considered as candidates for redescription set construction. The effects of changing minimal Jaccard index and minimal support size on the produced redescription set of size 50 by our framework on Country, Bio and DBLP dataset can be seen in Section S2.2.2 (Online resource 1).

We compared the CLUS-RM algorithm with the generalized redescription set construction procedure (CRM-GRS), to the ReReMi, the Split trees and the Layered trees algorithms implemented in the tool called Siren (Galbrun & Miettinen, 2012c). The specific parameter values used for each redescription mining algorithm can be seen in Section S2 (Online Resource 1).

5.3. Analysis of redescription sets produced with CRM-GRS

We analyse a set containing all redescriptions produced by CLUS-RM algorithm (referred to as a *large set of redescriptions*) and the corresponding sets of substantially smaller size constructed from this set by generalized redescription set construction procedure (referred to as *reduced sets of redescriptions*) on three different datasets.

For the purpose of this analysis, we create redescriptions without using the refinement procedure and disallow multiple redescriptions describing the same set of instances. To explore the influence of using different importance weights on properties of produced redescription sets, we use the different weight combinations given in Table 4.

Table 4: A matrix containing different combinations of importance weights for the individual redescription quality criteria.

$$W = \left[\begin{array}{cccccc} \mathrm{J} & \mathrm{pV} & \mathrm{AJ} & \mathrm{EJ} & RQS & RV \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.0 \\ 0.4 & 0.2 & 0.1 & 0.1 & 0.2 & 0.0 \\ 0.6 & 0.2 & 0.0 & 0.0 & 0.2 & 0.0 \\ 0.0 & 0.2 & 0.3 & 0.3 & 0.2 & 0.0 \end{array}\right]$$

In the rows $1, 2$ and 3 of matrix W, we incrementally increase the importance weight for the Jaccard index and equally decrease the weight for the element and attribute Jaccard index in order to explore the effects of finding highly accurate redescriptions at the expense of diversity. The last row explores the opposite setting that completely disregards accuracy and concentrates on diversity.

By using importance weights in each row of matrices W (Table 4) and W_{miss} (Table 5), we create redescription sets containing 25, 50, 75, 100, 125, 150, 175 and 200 redescriptions. We plot the change in element/attribute coverage, average redescription Jaccard index, average p-value, average element/attribute Jaccard index and average query size against the redescription set size. Information about redescriptions in the large set is used as a baseline and compared to the quality of reduced sets.

5.3.1. The analysis on the Bio dataset

We start the analysis by examining the properties of the large redescription set presented in Figure 5. In Figure 6, we compare the properties of redescriptions in the large redescription set, against properties of redescriptions in reduced sets based on different preference vectors. The results are presented only for the Bio dataset, however similar analysis for the DBLP and the Country dataset is presented in Section S2.2.3 (Online Resource 1).

Figure 5 shows distributions of quality measures for redescriptions in the large redescription set constructed with CLUS-RM algorithm. Redescription Jaccard index is mostly in $[0.6, 0.7]$ interval, though a noticeable number is in $[0.9, 1.0]$. The p-value is at most 0.01 but mainly smaller than 10^{-17}. The maximum average element Jaccard index equals 0.13 and the maximum average attribute Jaccard index equals 0.14 which shows a fair level of diversity among produced redescriptions. Over 99% of redescriptions contain less than 15 attributes in both queries, and more than 50% contains less than 10 attributes in both queries which is good for understandability.

Plots in Figure 6 contain 5 graphs demonstrating a specific property of the reduced redescription set and its change with the increase of reduced redescription set size. The *Reduced k* graph demonstrates properties of redescriptions contained in redescription set created with the preference weights from the k-th row of W. The graph labelled *Large set* demonstrates properties of redescriptions from a redescription set containing all produced redescriptions.

Increasing the importance weight for a redescription Jaccard index has the desired effect on redescription accuracy in the reduced sets of various size. Large weight on this criteria leads to sets with many highly accurate but more redundant redescriptions (average element Jaccard > 0.15) with larger support (average support $> 10\%$ of the total number of elements in the dataset). Consequence of larger support is increased overall element coverage. The effect is in part the consequence of using the Bio dataset that contains a number of accurate redescriptions with high support (also discussed in (Galbrun, 2013)). This effect is not observed on the Country and the DBLP dataset (Figures S4 and S5), where element and attribute coverage is increased only with increasing diversity weights in the preference vector. The average redescription Jaccard index decreases as the reduced set size increases which is expected since the total number of redescriptions with the highest possible accuracy is mostly smaller than 200.

Use of weights from the second row of the importance matrix W largely reduces redundancy and moderately lowers redescription accuracy in produced redescription set

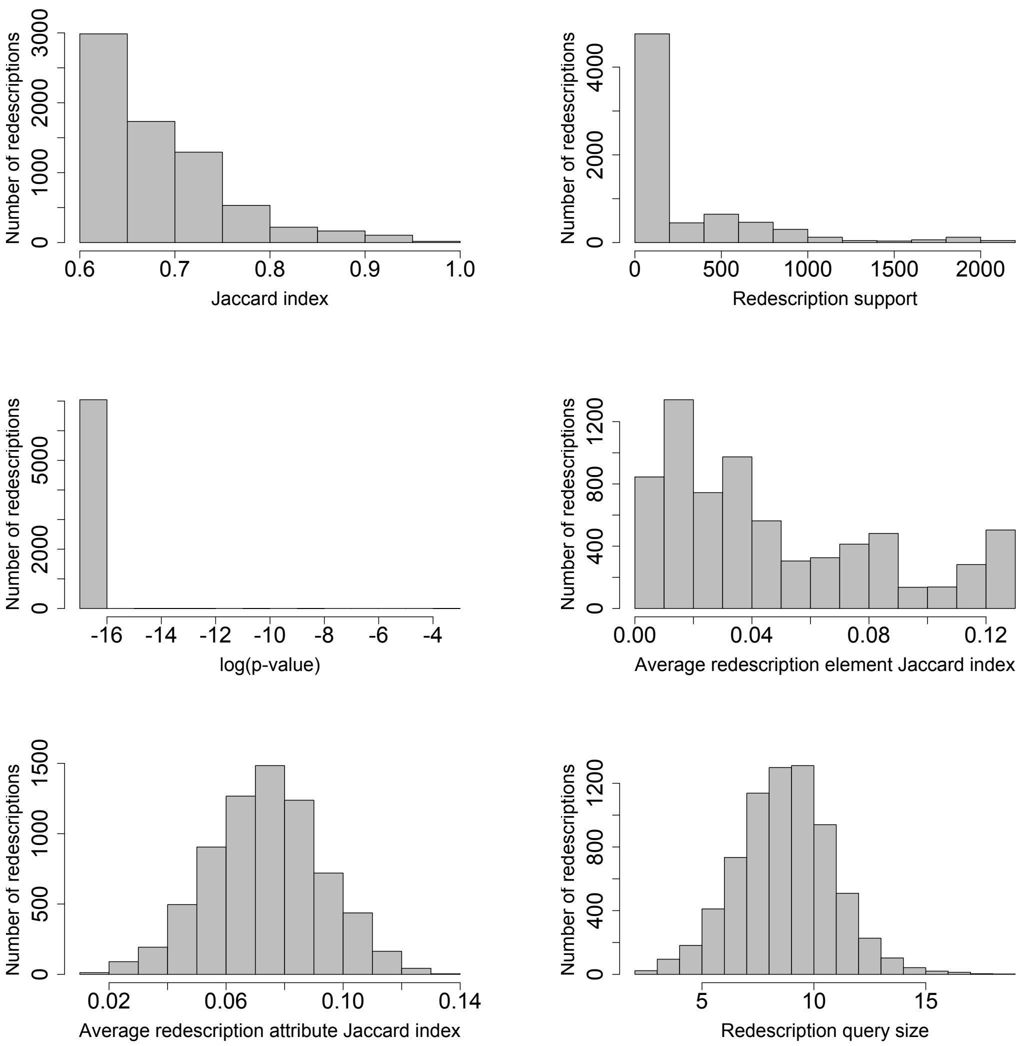


Figure 5: Histograms showing distributions of different redescription quality measures for the large redescription set containing 7413 redescriptions. Redescriptions are created on the Bio dataset.

compared to weights that highly favour redescription accuracy. The equal weight combination provides accurate redescriptions (above large set average) that describe different subsets of elements by using different attributes (both below large set average). The average redescription support is lower as a result, around 5% of data elements. Despite this, the element coverage is between 88% and 100% with the sharp increase to 98% for a set containing 50 redescriptions. The element coverage reaches 100% for sets containing at least 175 redescriptions.

Depending on the application, it might be interesting to find different, highly accurate descriptions of the same or very similar sets of elements (thus the weights from the third row of W from Table 4 would be applied). Higher redundancy provides different characteristics that define the group. It sometimes also provides more specific information about subsets of elements of a given group.

We found several highly accurate redescriptions describing very similar subsets of locations on the Bio dataset by using weights from the third row of the matrix W. These locations are characterized as a co-habitat of the Arctic fox and one of several other animals with some specific climate conditions. We provide two redescriptions describing a co-habitat of the Arctic fox and the Wood mouse.

$q_1 : \; -9.5 \le t_{11}^{-} \le 0.9 \;\wedge\; 9.7 \le t_7^{+} \le 13.4$

$q_2 : \;$ Woodmouse $\wedge$ ArcticFox $\wedge$ $\neg$ MountainHare

This redescription describes 57 locations with Jaccard index 0.83. One very similar redescription describing 58 lo-

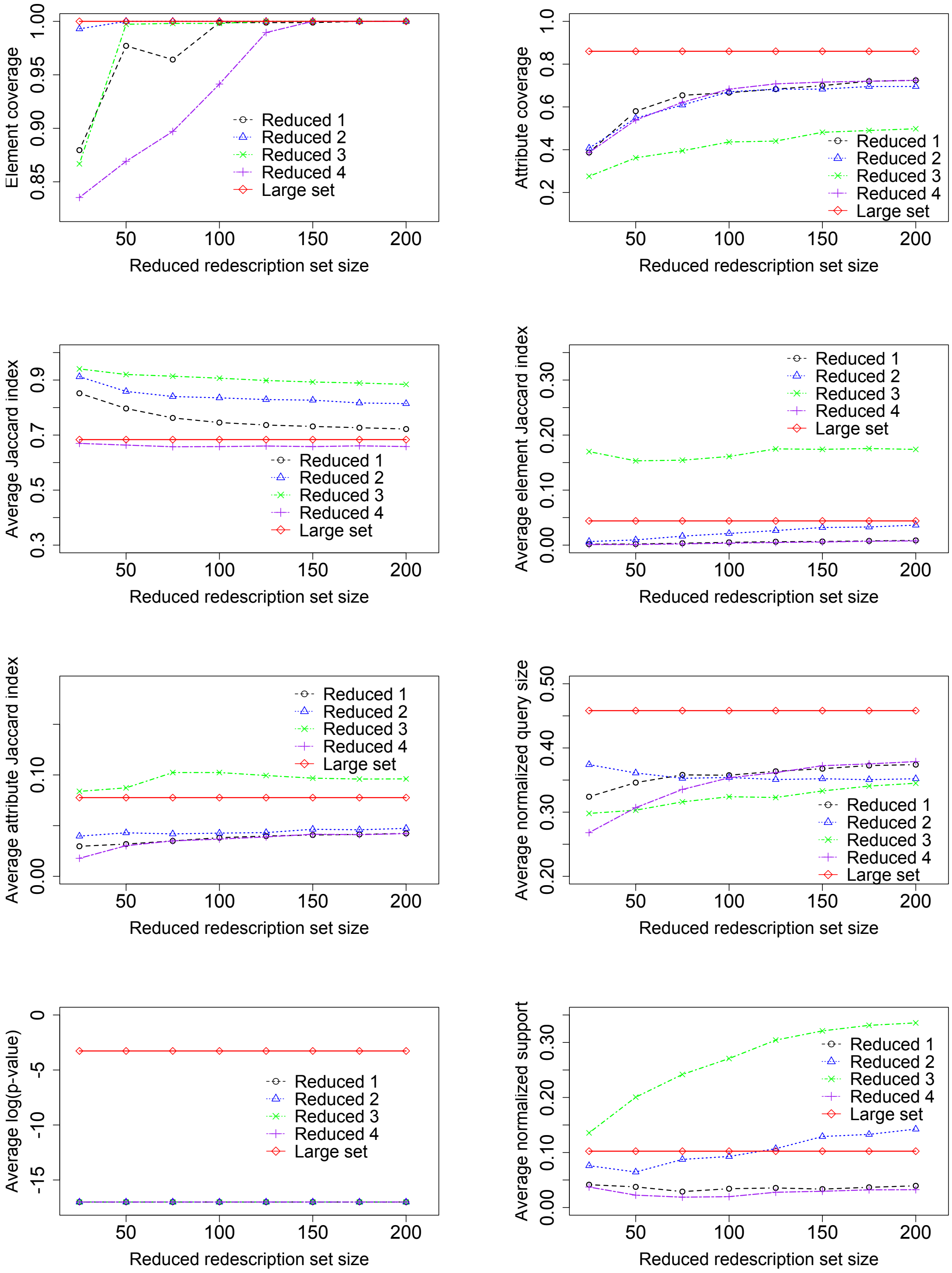


Figure 6: Plots comparing element and attribute coverage, average redescription: Jaccard index, log(p-value), element/attribute Jaccard index, normalized support and normalized query size for resulting reduced sets of different size and the original, large redescription set containing all produced redescriptions. *Reduced* k, corresponds to the reduced set obtained with the importance weights from the k-th row of the weight matrix W.

cations from which 57 are the same as above, with Jaccard index 0.87 is:
$q_1: \; -5.5 \leq \tilde{t}_2 \leq 2.2 \;\wedge\; 6.4 \leq t_9^+ \leq 10.6$
$q_2:$ Woodmouse $\wedge$ ArcticFox $\wedge$ $\neg$ Norwaylemming

Examples that are even more interesting can be found on the Country data where very similar sets of countries can be described by using different trading and general country properties. The example can be seen in Section S2.1.3, Figure S11 (Online Resource 1).

5.3.2. Using the redescription variability index on the Country dataset

We analyse the impact of missing values to redescription creation and use newly defined redescription variability index (RW), in the context of generalized set generation, on the Country dataset with a weight matrix shown in Table 5. The variability weight is gradually increased while other weights are equally decreased to keep the sum equal to 1.0 (which is convenient for interpretation).

Table 5: The weight matrix designed to explore the effects of changing redescription variability index on the resulting redescription set. These weights are applied on data containing missing values. Otherwise, the variability index weight (RV) should equal 0.

$$W_{miss} = \left[\begin{array}{cccccc} \text{J} & \text{pV} & \text{AJ} & \text{EJ} & RQS & RV \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.19 & 0.01 \\ 0.18 & 0.18 & 0.18 & 0.18 & 0.18 & 0.1 \\ 0.14 & 0.14 & 0.14 & 0.14 & 0.14 & 0.3 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.5 \\ 0.06 & 0.06 & 0.06 & 0.06 & 0.06 & 0.7 \end{array}\right]$$

The change in variability index depending on a reduced set size and comparison with the large set can be seen in Figure 7.

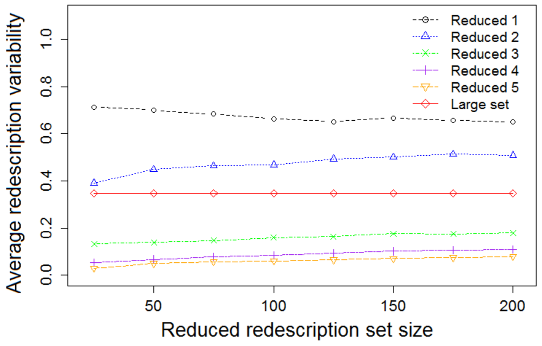


Figure 7: Change in average variability index of redescriptions in reduced redescription set for various set sizes and the set containing all created redescriptions.

As expected, increasing the importance weight for redescription variability favours selecting more stable redescriptions to the changes in missing values.

To demonstrate the effects of variability index to redescription accuracy, we plot graphs comparing averages of optimistic, query non-missing and pessimistic Jaccard index for every row of the weight matrix for different reduced set sizes. The results for row 1 and row 4 can be seen in Figures 8 and 9. Plots for reduced sets obtained with importance weights from the 2., the 3. and the 5. row of W_{miss} are available in Figure S12 (Online resource 1).

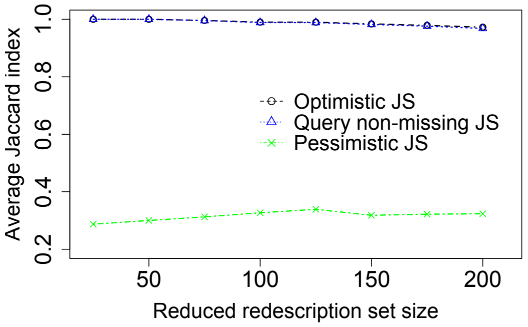


Figure 8: Optimistic, query non-missing and pessimistic Jaccard index for reduced sets of different sizes created with importance weight from the first row of the weight matrix W_{miss}.

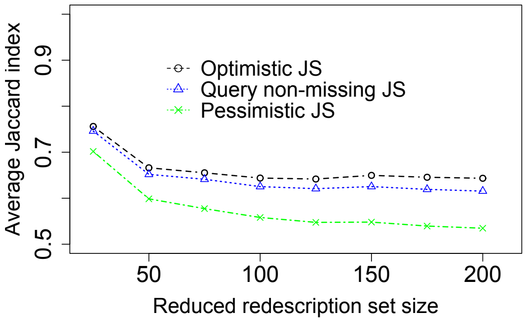


Figure 9: Optimistic, query non-missing and pessimistic Jaccard index for reduced sets of different sizes created with importance weight from the fourth row of the weight matrix W_{miss}.

Increasing the weight on the variability index has the desired effect of reducing the difference between values of different Jaccard index measures. However, the average optimistic and query non-missing Jaccard index values in the reduced sets drop as a result.
Redescription with $J_{qnm} = J_{pess} = J_{opt} = 1.0$:
$q_1: \; 3.6 \leq MORT \leq 4.1 \;\wedge\; 25.9 \leq \text{RUR\_POP} \leq 38.4$
$\wedge\; 58.8 \leq$ LABOR_PARTICIP_RATE ≤ 61.1
$q_2: \; 68.0 \leq E_{23} \leq 79.0 \;\wedge\; 0.7 \leq E/I_{104} \leq 4.4$
$\wedge\; 0.9 \leq E/I_{50} \leq 1.5$
is highly accurate and stable redescription constructed by

CRM-GRS with the importance weight from the fourth row of a matrix W_{miss}. It is statistically significant with the p-value smaller than 10^{-17}.

Redescriptions exist for which $J_{qnm} = J_{opt}$ and $J_{pess} < J_{opt}$. In such cases, the drop in accuracy from J_{opt} to J_{pess} occurs because a number of elements exist in the dataset for which membership in the support of neither redescription query can be determined, due to missing values. Optimizing pessimistic Jaccard index is very strict and can discard some potentially significant redescriptions such as:
$q_1 :\ 5.6 \leq$ EMPL_BAD $\leq 18.2\ \wedge\ 2.9 \leq MORT \leq 4.5$ $\wedge\ 2.0 \leq$ AGR_EMP $\leq 10.5\ \wedge\ -2.4 \leq$ BAL ≤ 10.1
$q_2 :\ 1.1 \leq E/I_{85} \leq 3.1\ \wedge\ 93.0 \leq E_{97} \leq 98.0$. This redescription has $J_{qnm} = J_{opt} = 1.0$ and $J_{pess} = 0.48$. With the variability index of 0.52 it describes all elements that can be evaluated by at least one redescription query with the highest possible accuracy.

This example motivates optimizing query non-missing Jaccard with positive weight on the variability index. It is especially useful when small number of highly accurate redescriptions can be found and when a large percentage of missing values is present in the data.

5.4. Evaluating the conjunctive refinement procedure

The next step is to evaluate the conjunctive refinement procedure and its effects on the overall redescription accuracy. We use the same experimental set-up as in Section 5.3 for both sets with the addition of the minimum refinement Jaccard index parameter, which was set to 0.4 on the Bio dataset and 0.1 on the Country and the DBLP dataset. The algorithm requires the initial clusters to start the mining process as explained in Section 3.1.1 and in (Mihelčić et al., 2015b). To maintain the initial conditions, we create one set of initial clusters and use them to create redescriptions with and without the conjunctive refinement procedure. Since we use PCTs with the same initial random generator seed in both experiments, the differences between sets are the result of applying the conjunctive refinement procedure. The effects of using conjunctive refinement are examined on sets containing all redescriptions produced by CLUS-RM and on reduced sets created with equal importance weights by the generalized redescription set construction procedure (Row 1 in matrix W).

The effects of using the refinement procedure on redescription accuracy are demonstrated in comparative histogram (Figure 10) showing the distribution of redescription Jaccard index in a set created by CLUS-RM with and without the refinement procedure.

CLUS-RM produced 7413 redescriptions, satisfying constraints from Section 5.2, without the refinement procedure and 10472 redescriptions with the refinement procedure. The substantial increase in redescriptions satisfying user-defined constraints, when the conjunctive refinement procedure is used, is accompanied by significant improvement in redescription accuracy.

We performed the one-sided independent 2-group Mann-Whitney U test with the null hypothesis that there is a probability of 0.5 that an arbitrary redescription (R_r) from a set obtained by using conjunctive refinement has the Jaccard index larger than the arbitrary redescription (R_{nr}) from a set obtained without using the conjunctive refinement procedure ($P(J(R_r) > J(R_{nr})) = 0.5$). The p-value of $2.2 \cdot 10^{-16}$ lead us to reject the null hypothesis with the level of significance 0.01 and conclude that $P(J(R_r) > J(R_{nr})) > 0.5$ must be true.

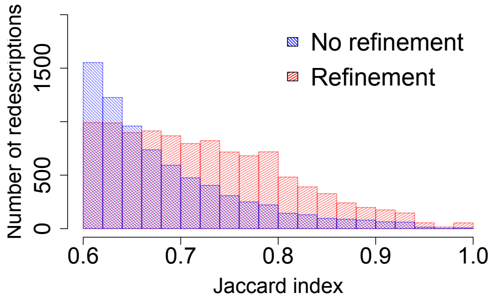


Figure 10: Distribution of a redescription Jaccard index in a large set created on a Bio dataset with and without the conjunctive refinement procedure. The set obtained without using the conjunctive refinement procedure contains 7413 redescriptions, and the set obtained by using the conjunctive refinement procedure contains 10472 redescriptions.

Another useful property of the conjunctive refinement procedure is that it preserves the size of redescription support. The comparative distribution of redescription supports between the sets is shown in Figure 11.

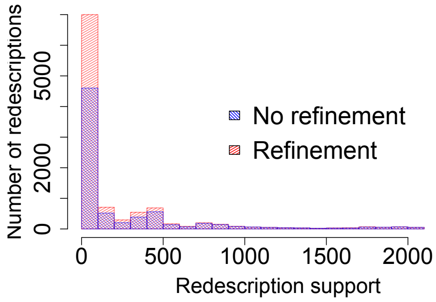


Figure 11: Distribution of a redescription support size in a large set created on a Bio dataset with and without the conjunctive refinement procedure. The set obtained without using the conjunctive refinement procedure contains 7413 redescriptions, and the set obtained by using the conjunctive refinement procedure contains 10472 redescriptions.

Majority of 3059 redescriptions that entered the redescription set because of the improvements made by the conjunctive refinement have supports in the interval $[10, 500]$ elements. Because of that, the average support size in the redescription set obtained by using the refinement procedure (217.98) is lower than that obtained without the refinement procedure (263.63). The change in distribution is significant, as shown by the one-sided independent 2-group Mann-Whitney U test. The test rejects the hypothesis $P(|supp(R_{nr})| > |supp(R_r)|) = 0.5$ with the level of significance 0.01 (p-value equals $2.4 \cdot 10^{-14}$), thus showing that $P(|supp(R_{nr})| > |supp(R_r)|) > 0.5$.

Using the conjunctive refinement procedure improves redescription accuracy and adds many new redescriptions to the redescription set. However, since the reduced sets are presented to the user, it is important to see if higher quality reduced sets can be created from the large set by using the conjunctive refinement procedure compared to the set obtained without using the procedure.

We plot comparative distributions for all defined redescription measures for reduced sets extracted from the redescription set obtained with (*CLRef*) and without (*CLNRef*) the conjunctive refinement procedure. The comparison made on the sets containing 200 redescriptions is presented in Figure 12. The boxplots representing distributions of supports show that the redescription construction procedure extracts redescriptions of various support sizes, which was intended to prevent focusing only on large or small redescriptions based on redescription accuracy.

We compute the one-sided independent 2-group Mann-Whitney U test on the reduced sets for the redescription Jaccard index (J) and the normalized redescription query size (RQS) since there seem to be a difference in distributions as observed from Figure 12. For other measures, we compute the two-sided Mann-Whitney U test to assess if there is any notable difference in values between the sets.

The null hypothesis that $P(J(R_r) > J(R_{nr})) = 0.5$ is rejected with the p-value smaller than $2.2 \cdot 10^{-16} < 0.01$, thus the alternative hypothesis $P(J(R_r) > J(R_{nr})) > 0.5$ holds. The difference in support between two sets is not statistically significant (p-value equals 0.21, obtained with the two-sided test). Distributions of redescription p-values are identical because all redescriptions have equal p-value: 0.0. The difference in average attribute/element Jaccard index is also not statistically significant (p-values 0.88 and 0.13 respectively obtained with the two-sided test). The p-value for the null hypothesis $P(RQS(R_{nr}) < RQS(R_r)) = 0.5$ equals $5.25 \cdot 10^{-6} < 0.01$ thus the alternative hypothesis $P(RQS(R_{nr}) < RQS(R_r)) > 0.5$ holds.

The refinement procedure enables constructing reduced sets containing more accurate redescriptions with the average Jaccard index increasing from 0.72, for reduced set obtained without using refinement procedure, to 0.82 for reduced set obtained when refinement procedure is used. This improvement sometimes increases redescription complexity, albeit this is limited on average to having less than 1 additional attribute in redescription queries.

The set produced by using the conjunctive refinement procedure has the element coverage of 0.9996 and the attribute coverage of 0.7613 compared to the set where this procedure was not used where the element coverage is 1.0 and the attribute coverage is 0.7243.

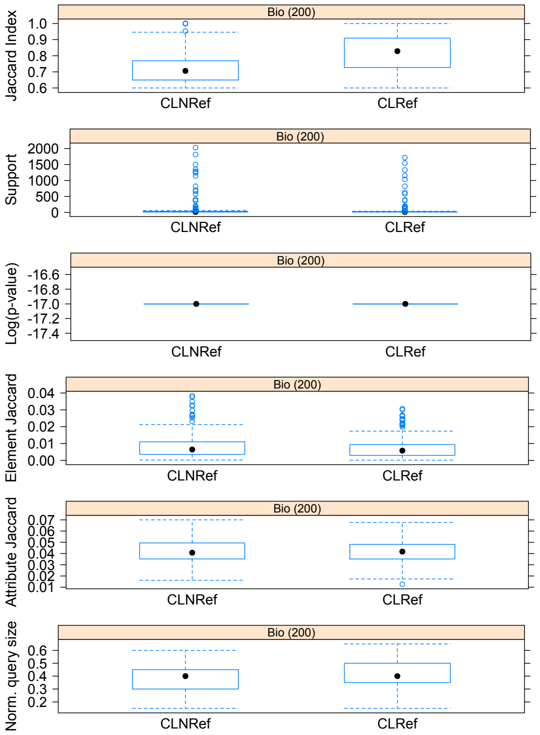


Figure 12: Boxplots comparing distributions of redescription: Jaccard index, support, log(p-value), element Jaccard index, attribute Jaccard index and normalized query size in reduced sets containing 200 redescriptions. The reduced sets were obtained by the generalized redescription set construction procedure by using equal importance weight for each measure.

The conjunctive refinement procedure also significantly increases redescription accuracy on the DBLP and the Country dataset. Equivalent analysis for these datasets is performed in Section S2.3 (Online Resource 1).

5.5. Comparisons with other state of the art redescription mining algorithms.

In this section, we present the comparative results of redescription set quality produced by our framework (CRM-GRS) compared to the state of the art algorithms: the ReReMi Galbrun & Miettinen (2012b), the Split trees and the Layered trees Zinchenko (2014). To perform the experiments, we used the implementation of the ReReMi, the Split trees and the Layered trees algorithm within the tool Siren (Galbrun & Miettinen, 2012c).

The ReReMi algorithm was already compared in (Galbrun & Miettinen, 2012b) with the CartWheels algorithm (Ramakrishnan et al., 2004) (on a smaller version of a DBLP and the Bio dataset), with the association rule mining approach obtained by the ECLAT frequent itemset miner (Zaki, 2000) and the greedy approach developed by Gallo et al. (2008). The approach from Zaki & Ramakrishnan (2005), which is also related, works only with boolean attributes and have no built in mechanism to differentiate different views. Redescription mining on the DBLP dataset with the original implementation of the algorithm[1] returned 49 redescriptions, however they only describe authors by using co-authorship network. Since, our goal is to describe authors by their co-authorship network and provide the information about the conferences they have published in, these redescriptions are not used in our evaluation. To use the approach on the Bio dataset, we first applied the *Discretize* filter in weka[2] to obtain nominal attributes. Then, we applied *NominalToBinary* filter to obtain binary attributes that can be used in Charm-L. As a result, the number of attributes on the Bio dataset increased to 1679 making the process of constructing a lattice of closed itemsets to demanding with respect to execution time constraints. The Country dataset contains missing values which are not supported by this approach.

Since there is an inherent difference in the number of created redescriptions, depending on the type of logical operators used to create them, between CLUS-RM and the comparative algorithms, we split the algorithm comparison in two parts. First, we compare redescription properties created by using all logical operators and then redescriptions created by using only the conjunction and the negation operator (Bio and DBLP dataset) or only by using the conjunction operator (Country dataset).

After obtaining redescriptions with the algorithms implemented in the tool Siren (Galbrun & Miettinen, 2012c), with parameters specified in Section 5.2, we used the *Filter redundant redescriptions* option to remove duplicate and redundant redescriptions. Since SplitTrees and LayeredTrees algorithms always use all logical operators to create redescriptions, we created a redescription set with these approaches and filtered out redescriptions containing the disjunction operator in at least one of its queries.

For each obtained redescription set from the ReReMi, the Split trees and the Layered trees algorithm, we extracted a redescription set of the same size with the generalized redescription set procedure with equal weight importance for each redescription criteria. These sets are extracted from a large set created with the CLUS-RM algorithm with the parameters specified in Section 5.2.

We plot pairwise comparison boxplots for each redescription measure comparing the performance of our framework with the three chosen approaches.

For each comparison we analyse the hypothesis about the distributions by using the one-sided independent 2-group Mann-Whitney U test (see summary in Table 6).

5.5.1. Comparison on the Bio dataset

First, we compare the algorithms on the Bio dataset. Figures 13, 14 and Table 6 show that the set produced by CRM-GRS tend to contain more accurate redescriptions on the Bio dataset when the conjunction and the negation operators are allowed and when the conjunctive refinement procedure is used compared to all other approaches.

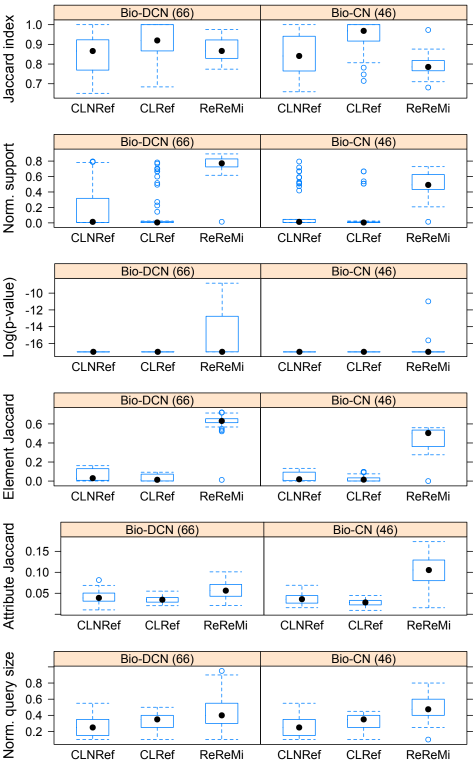


Figure 13: Boxplots comparing redescriptions produced with our framework (CLNref, CLRef) and the ReReMi algorithm (ReReMi) on the Bio dataset. Sets contain 66 redescriptions created by using all defined logical operators and 46 redescriptions when only conjunction and negation operators are used to construct redescription queries.

The results are significant at the significance level of 0.01, except for the case of ReReMi when all logical operators were allowed and refinement procedure was not used in the CLUS-RM algorithm. Redescriptions contained in redescription sets produced by CRM-GRS tend to have smaller p-values compared to redescriptions produced by other tree - based algorithms (statistically significant with the significance level of 0.05). Redescription sets created by CRM-GRS tend to contain redescriptions with smaller element/attribute Jaccard index (redundancy) and smaller query size (the difference is statistically significant with the significance level of 0.01 with the exception of a set created by CRM-GRS, when conjunctive refinement procedure was not used in CLUS-RM, compared to the set created by Layered trees algorithm).

Element and attribute coverage analysis for all approaches is provided in Section S2.5.1 (Online Resource 1). This analysis suggests that despite smaller average

[1] http://www.cs.rpi.edu/~zaki/www-new/pmwiki.php/Software/Software

[2] http://www.cs.waikato.ac.nz/ml/weka/

Table 6: Table containing p-values obtained with the one-sided independent 2-group Mann-Whitney U test. We test the hypothesis to have the probability 0.5 that the redescription chosen from the redescription set obtained by our framework has larger/smaller value compared to the redescription chosen from the redescription set produced by the ReReMi, the Split trees (ST) or the Layered trees (LT), depending on the redescription measure used, compared to the alternative in which a redescription chosen from a set produced by our framework has the probability greater than 0.5 for this outcome. For the Jaccard index (J) and support we test if the probability is greater than 0.5 to obtain larger values, for the average redescription redundancy based on elements/attributes contained in their support (AEJ)/ (AAJ) and redescription query size (RQS), we test if the probability is larger to obtain smaller values in the set produced by our framework. Each table cell contains two p-values in the format $pVal1/pVal2$. The first p-value relates to the set produced by the CLUS-RM without the conjunctive refinement procedure and the second with the refinement procedure.

Dataset	Operators	Measure	ReReMi	ST	LT
Bio	AllOp (DCN)	J	$0.91/2 \cdot 10^{-4}$	$2.6 \cdot 10^{-9}/2.7 \cdot 10^{-15}$	$0.0035/1.9 \cdot 10^{-7}$
		Supp	1.0/1.0	1.0/1.0	0.9994/1.0
		p-value	$2 \cdot 10^{-9}/2 \cdot 10^{-9}$	0.0217/0.0217	0.0408/0.0408
		AEJ	$< 2 \cdot 10^{-16}/< 2 \cdot 10^{-16}$	$5.3 \cdot 10^{-10}/3.4 \cdot 10^{-11}$	$1.3 \cdot 10^{-5}/2.3 \cdot 10^{-8}$
		AAJ	$2 \cdot 10^{-7}/2 \cdot 10^{-7}$	$1.2 \cdot 10^{-13}/< 2 \cdot 10^{-16}$	$0.1122/8.2 \cdot 10^{-5}$
		RQS	$2 \cdot 10^{-8}/9 \cdot 10^{-5}$	$1.5 \cdot 10^{-8}/1.3 \cdot 10^{-5}$	$6.7 \cdot 10^{-7}/5 \cdot 10^{-5}$
	ConjNeg (CN)	J	$0.0035/1.5 \cdot 10^{-12}$	$\vert\mathcal{R}\vert < 10$	$\vert\mathcal{R}\vert < 10$
		Supp	1.0/1.0		
		p-value	0.08/0.08		
		AEJ	$< 2 \cdot 10^{-16}/1.4 \cdot 10^{-15}$		
		AAJ	$< 2 \cdot 10^{-16}/< 2 \cdot 10^{-16}$		
		RQS	$4.3 \cdot 10^{-10}/3.5 \cdot 10^{-7}$		
DBLP	AllOp (DCN)	J	1.0/1.0	1.0/0.9999	$\vert\mathcal{R}\vert < 10$
		Supp	1.0/1.0	0.0033/0.0033	
		p-value	1.0/1.0	1.0/1.0	
		AEJ	1.0/1.0	0.904/0.980	
		AAJ	1.0/1.0	0.9997/0.9998	
		RQS	$< 2 \cdot 10^{-16}/8.6 \cdot 10^{-9}$	$< 2 \cdot 10^{-16}/3.5 \cdot 10^{-15}$	
	ConjNeg (CN)	J	$0.0127/5.96 \cdot 10^{-7}$	$\vert\mathcal{R}\vert < 10$	$\vert\mathcal{R}\vert < 10$
		Supp	$1.74 \cdot 10^{-8}/1.14 \cdot 10^{-9}$		
		p-value	0.9779/0.9933		
		AEJ	1.0/1.0		
		AAJ	1.0/1.0		
		RQS	1.0/1.0		
Country	AllOp (DCN)	J_{pess}	1.0/0.9979		
		J_{qnm}	$< 2 \cdot 10^{-16}/< 2 \cdot 10^{-16}$		
		Supp	1.0/1.0		
		p-value	$6.3 \cdot 10^{-10}/7.5 \cdot 10^{-10}$	NA	NA
		AEJ	$< 2 \cdot 10^{-16}/< 2 \cdot 10^{-16}$	NA	NA
		AAJ	$< 2 \cdot 10^{-16}/< 2 \cdot 10^{-16}$		
		RQS	$< 2 \cdot 10^{-16}/< 2 \cdot 10^{-16}$		
	Conj (CN)	J_{pess}	$0.257/7 \cdot 10^{-6}$		
		J_{qnm}	$5.2 \cdot 10^{-7}/2.3 \cdot 10^{-8}$		
		Supp	$4.7 \cdot 10^{-4}/0.769$		
		p-value	0.0503/0.0239	NA	NA
		AEJ	$0.608/2.6 \cdot 10^{-5}$	NA	NA
		AAJ	$1.74 \cdot 10^{-15}/3.3 \cdot 10^{-12}$		
		RQS	$1.3 \cdot 10^{-9}/3.7 \cdot 10^{-17}$		

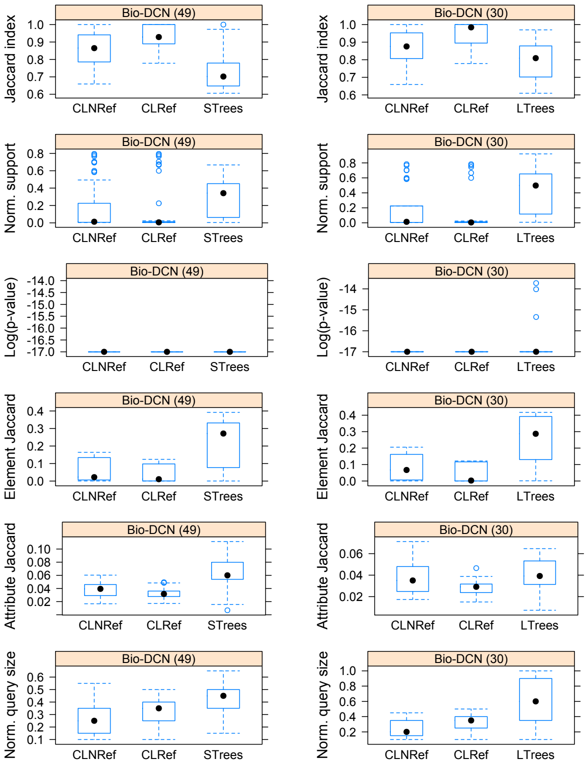


Figure 14: Boxplots comparing 49 redescriptions by using all defined logical operators, produced with our framework (CLNref, CLRef) and the Split trees algorithm (STrees) on the Bio dataset (left). The analogous comparison is made with the Layered trees algorithm (LTrees) on 30 redescriptions (right).

redescription support, our framework has comparable performance with respect to element and attribute coverage.

As already discussed in (Galbrun, 2013), the ReReMi algorithm has a drift towards redescriptions with large supports on the Bio dataset. The consequence is a large element redundancy among produced redescriptions. The Split trees and the Layered trees algorithms produce redescriptions in the whole support range, though majority of produced redescriptions still have a very high support resulting in large element redundancy. Our approach returns redescriptions with various support size as can be seen from Figures 13 and 14 though majority of produced redescriptions are very close to the minimal allowed support. However, if needed, the minimal support can be adjusted to produce sets containing redescriptions that describe larger sets of elements. It is also possible to produce multiple sets, each being produced with different minimal and maximal support bounds. Also, by adjusting the importance weights to highly favour Jaccard index, the user can produce reduced sets with similar properties as those produced by the ReReMi, the Layered trees and the Split trees. The distribution of support size in the large redescription set produced with the basic variant of CLUS-RM algorithm on the Bio dataset can be seen in Figure 5. The increase in accuracy obtainable by using different weights to construct reduced sets can be seen in Figure 6.

Redescription sets produced with the Layered and the Split trees algorithms do not create enough redescriptions containing only conjunction and negation operator in its queries to make the distribution analysis. The Layered trees algorithm produced only one redescription with Jaccard index 0.62 and the Split trees algorithm created four redescriptions with Jaccard index 0.97, 0.65, 0.7 and 0.78. On the other hand, the CLUS-RM with the conjunctive refinement procedure created over 14000 redescriptions containing only conjunction and negation in the queries with the Jaccard index greater than 0.6 from which 73 redescriptions have Jaccard index 1.0.

Our framework complements the existing approaches which is visible from redescription examples found by our approach that were not discovered by other algorithms. Section S2.5.1 (Online Resource 1) contains one example of very similar redescription, found by the ReReMi and the CRM-GRS, and several redescriptions discovered by CRM-GRS that were not found by other approaches.

5.5.2. Comparison on the DBLP dataset

The DBLP dataset is very sparse and all redescription mining algorithms we tested only returned a very small number of highly accurate redescriptions. Half of the redescription mining runs we performed with different algorithms returned to small number of redescriptions to perform a statistical analysis. On this dataset, we can compare quality measure distributions of redescriptions produced by our framework only with the ReReMi algorithm (Figure 15), and with the Split trees algorithm when all operators are used to construct redescription queries. (Figure 16). CRM-GRS tends to produce redescriptions with smaller query size than the ReReMi and the Split trees algorithms when all the operators are allowed. The redescriptions contained in the reduced set produced by our framework tend to have higher support than those produced by the Split trees algorithm. The distribution analysis on sets created by using only conjunction and negation logical operators can be performed only against the ReReMi algorithm due to small number of redescriptions produced by the other approaches. In this case, CRM-GRS tends to produce more accurate redescriptions (significant at the significance level of 0.01 when the conjunctive refinement is used and at the significance level of 0.05 when conjunctive refinement is not used). In both cases, our framework produces redescriptions that tend to have larger support (significant with the level of 0.01). There is a more pronounced difference between the Split trees algorithm and CRM-GRS when all the operators are allowed. In this case, the Split trees algorithm has higher median in distribution of redescription accuracy.

The Layered trees approach produced 7 redescriptions using all operators, with accuracy

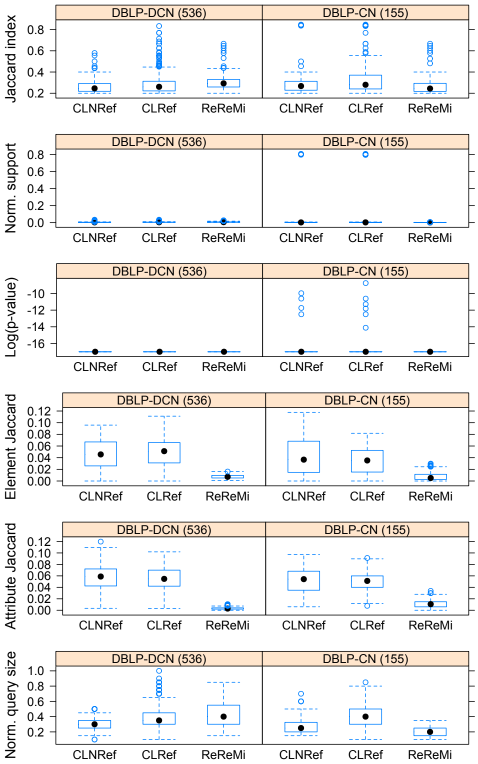


Figure 15: Boxplots comparing redescriptions, produced with our framework (CLNref, CLRef) and the ReReMi algorithm (ReReMi) on the DBLP dataset. Sets contain 536 redescriptions created by using all defined logical operators and 155 redescriptions when only conjunction and negation operators are used to construct redescription queries.

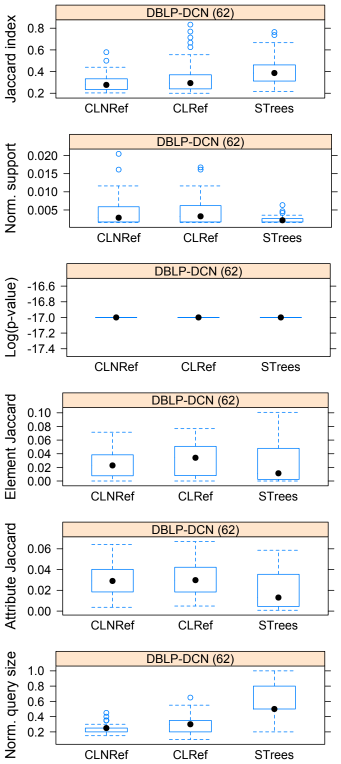


Figure 16: Boxplots comparing redescriptions, produced with our framework (CLNref, CLRef) and the Split trees algorithm (STrees) on the DBLP dataset. The set contains 62 redescriptions created by using all defined logical operators.

$0.85, 0.81, 0.71, 0.73, 0.23, 0.23, 0.2$ describing 10 to 48 authors. It produced 3 redescriptions using only conjunction and negation operators. The produced redescriptions had the accuracy $0.23, 0.22, 0.2$ and the support 45 to 48 authors. The Split trees algorithm produced only one redescription with accuracy 0.33 and support 13 using only conjunction and negation operators.

The most accurate redescriptions produced by each algorithm and a short discussion can be seen in Section S2.5.2 (Online Resource 1).

5.5.3. Comparison on the Country dataset

Comparisons on the Country dataset are preformed only with the ReReMi algorithm since it is the only algorithm, besides CLUS-RM, that can work on datasets containing missing values. Techniques for value imputation must be used before other approaches can be applied. Using these techniques introduces errors in the descriptions and violates a property of descriptions being valid for each element in redescription support. Because of that, we chose not to pursue this line of research.

Since our framework optimizes the query non-missing Jaccard index and the ReReMi optimizes pessimistic Jaccard index, we decided to make comparisons using both measures (Figure 17 and Figure 18). We extract two sets with CRM-GRS, for each we use different Jaccard index as one of the quality criteria. Redescriptions produced by the ReReMi remain unchanged but we compute the query non-missing Jaccard for each redescription which causes redescription accuracy to rise. Optimizing pessimistic Jaccard seems like the best option for comparisons since then the query non-missing Jaccard index necessarily increases and the redescription support is preserved.

Results from Table 6 show that CRM-GRS produces redescription set that tends to contain more accurate redescriptions when conjunction refinement procedure is used. The result is significant at the significance level 0.01. However, it failed to produce such set using all operators when pessimistic Jaccard index is used to evaluate redescription accuracy (redescription set produced by ReReMi has higher median in accuracy). Although, CRM-GRS produced a few redescriptions with higher accuracy than those produced by the ReReMi. When query non-missing Jaccard index is used as accuracy evaluation cri-

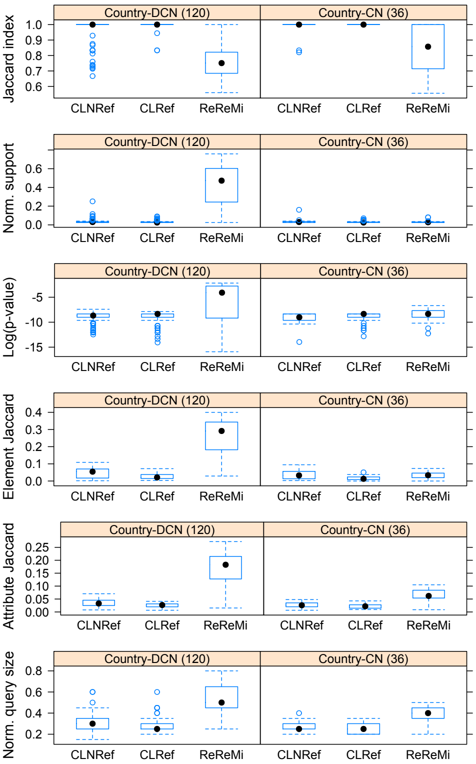


Figure 17: Boxplots comparing redescriptions, produced with our framework (CLNref, CLRef) and the ReReMi algorithm (ReReMi) on the Country dataset. Sets contain 120 redescriptions created by using all defined logical operators and 36 redescriptions when only conjunction and negation operators are used to construct redescription queries. Redescription accuracy is evaluated by using query non - missing Jaccard index.

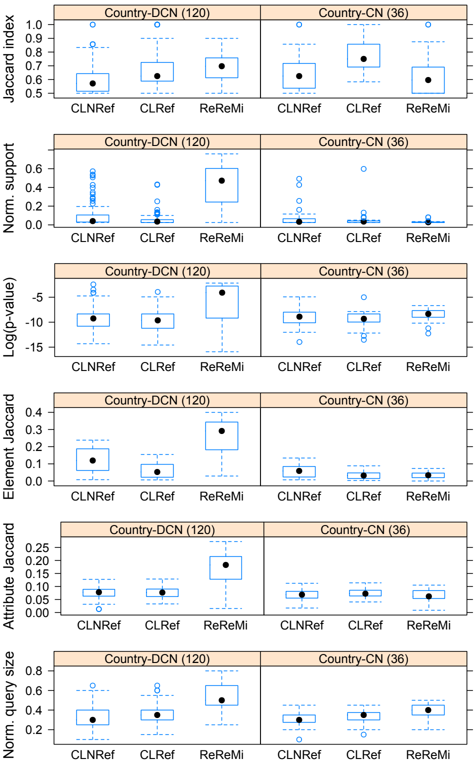


Figure 18: Boxplots comparing redescriptions, produced with our framework (CLNref, CLRef) and the ReReMi algorithm (ReReMi) on the Country dataset. Sets contain 120 redescriptions created by using all defined logical operators and 36 redescriptions when only conjunction and negation operators are used to construct redescription queries. Redescription accuracy is evaluated by using pessimistic Jaccard index.

teria, CRM-GRS tends to create more accurate redescriptions than the ReReMi (statistically significant at the significance level 0.01). When using only conjunction logical operator, the ReReMi tends to produce redescriptions with smaller support compared to CRM-GRS if conjunctive refinement procedure is not used.

Analysis of element and attribute coverage is provided in Section S2.5.3 (Online Resource 1).

The ReReMi algorithm found 2 redescriptions with $J_{pess} = 1.0$ while CRM-GRS created redescription set containing 4 redescriptions with $J_{pess} = 1.0$ when only conjunction operators are allowed and 5 redescriptions when all operators are allowed.

The analysis of comparative redescription examples produced by CRM-GRS and the ReReMi algorithm can be seen in Section S2.5.3 (Online Resource 1).

The ReReMi produced 14 redescriptions with $J_{qnm} = 1.0$ using only conjunction operators while redescription sets constructed by CRM-GRS contain 34 out of 36 redescriptions with $J_{qnm} = 1.0$ without using conjunctive refinement and 36 out of 36 redescriptions with $J_{qnm} = 1.0$ with the use of conjunctive refinement procedure. When all logical operators were used to create redescriptions, the ReReMi creates large number of disjunction based redescriptions, many of which are quite complex.

The difference in support size of redescriptions produced by CRM-GRS compared to those produced by the ReReMi algorithm, visible in Figures 17 and 18 when all operators are used is in part the consequence of CRM-GRS using high weight on element diversity but is also connected to different logic in using the disjunction operator. CRM-GRS allows improving Jaccard index, by using disjunctions, only for redescriptions satisfying a predefined accuracy threshold. Highly overlapping subsets of instances are thus complemented with subsets that are highly overlapping with one of the already existing subset of instances. Because of this, our framework eliminates descriptions of unrelated subsets of instances that occasionally occur in ReReMi's descriptions as a result of using disjunction operator (discussed in (Galbrun, 2013)).

6. Conclusions

We have presented a redescription mining framework CRM-GRS which integrates the generalized redescription set construction procedure with the CLUS-RM algorithm (Mihelčić et al., 2015a,b).

The main contribution of this work is the generalized redescription set construction procedure that allows creating multiple redescription sets of reduced size with different properties defined by the user. These properties are influenced by the user through importance weights on different redescription criteria. Use of the scalarization technique developed in multi - objective optimization guarantees that, at each step, one non-dominated redescription is added to the redescription set under construction. The generalized redescription set construction procedure has lower worst time complexity than existing redescription mining algorithms so it may be preferred choice over the multiple runs of these algorithms. The procedure allows creating sets of different size with different redescription properties. These features generally lack in current redescription mining approaches, where users are forced to experiment with individual algorithm parameters in order to obtain desirable set of redescriptions. Finally, the procedure allows using ensembles of redescription mining algorithms to create reduced sets with superior properties compared to those produced by individual algorithms.

The second contribution is related to increasing overall redescription accuracy. Here, we build upon our previous work on CLUS-RM algorithm and provide new - conjunctive refinement procedure, that significantly enlarges and improves the accuracy of redescriptions in the baseline redescription set by combining candidate redescriptions during the generation process. This procedure can be easily applied in the context of majority of other redescription mining algorithms, thus we consider it as a generally useful contribution to the field of redescription mining.

Finally, we motivate the use of query non-missing Jaccard index, introduced in (Mihelčić et al., 2015b), when data contains missing values. We show that using pessimistic Jaccard index eliminates some potentially useful, high quality redescriptions obtainable by using query non-missing Jaccard index. To further increase the possibilities of redescription mining algorithms, we introduce the redescription variability index that allows extracting stable redescriptions in the context of missing data, by combining the upper and lower bound on estimates of Jaccard index.

The evaluation of our framework with 3 different state of the art algorithms on 3 different real-world datasets shows that our framework significantly outperforms other approaches in redescription accuracy in majority of cases. In particular in settings when only conjunction and negation operators are used in redescriptions, which is the preferred setting from the point of understandability. In general, CRM-GRS produces more understandable redescriptions (due to smaller query size and extensive use of conjunction operator), it is more flexible and in majority of comparisons more accurate approach to mine redescriptions from datasets. Moreover, we demonstrated that it complements existing approaches in the discovered redescriptions and solves several problems of existing approaches (mainly the problem of support drift and redescriptions connecting unrelated parts of element space by using disjunctions). The framework is easily extendible with new redescription criteria and allows combining results of different redescription mining algorithms to create reduced sets with superior properties with respect to different redescription quality criteria.

Acknowledgement

The authors would like to acknowledge the European Commission's support through the MAESTRA project (Gr. no. 612944), the MULTIPLEX project (Gr.no. 317532), and support of the Croatian Science Foundation (Pr. no. 9623: Machine Learning Algorithms for Insightful Analysis of Complex Data Structures).

References

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., & Verkamo, A. I. (1996). Fast discovery of association rules. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 307–328). Menlo Park, CA, USA: American Association for Artificial Intelligence.

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

Bickel, S., & Scheffer, T. (2004). Multi-view clustering. In *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), 1-4 November 2004, Brighton, UK* (pp. 19–26). doi:10.1109/ICDM.2004.10095.

Blockeel, H., & De Raedt, L. (1998). Top-down induction of first-order logical decision trees. *Artif. Intell.*, *101*, 285–297. doi:10.1016/S0004-3702(98)00034-4.

Bouker, S., Saidi, R., Yahia, S. B., & Nguifo, E. M. (2012). Ranking and selecting association rules based on dominance relationship. In *Proceedings of the 24th IEEE International Conference on Tools with Artificial Intelligence, (ICTAI 2012), Athens, Greece, November 7-9, 2012* (pp. 658–665). doi:10.1109/ICTAI.2012.94.

Boulicaut, J.-F., & Bykowski, A. (2000). Frequent closures as a concise representation for binary data mining. In *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications (PAKDD 2000), Tokyo Japan* (pp. 62–73). Springer Berlin Heidelberg. doi:10.1007/3-540-45571-X_9.

Caramia, M., & Dell'Olmo, P. (2008). *Multi-objective Management in Freight Logistics: Increasing Capacity, Service Level and Safety with Optimization Algorithms*. Springer London.

Cox, D. R. (1957). Note on grouping. *Journal of the American Statistical Association*, *52*, 543–547.

DBLP (2010). DBLP dataset. URL: http://dblp.uni-trier.de/db accessed March 2010.

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 1999) (pp. 43–52). New York, NY, USA: ACM. doi:10.1145/312129.312191.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Mach. Learn.*, *2*, 139–172. doi:10.1023/A:1022852608280.

Fisher, W. D. (1958). On Grouping for Maximum Homogeneity. *Journal of the American Statistical Association*, *53*. doi:10.2307/2281952.

Galbrun, E. (2013). *Methods for Redescription Mining*. Ph.D. thesis University of Helsinki Finland.

Galbrun, E., & Kimmig, A. (2013). Finding relational redescriptions. *Machine Learning*, *96*, 225–248. doi:10.1007/s10994-013-5402-3.

Galbrun, E., & Miettinen, P. (2012a). A case of visual and interactive data analysis: Geospatial redescription mining. In *Instant Interactive Data Mining Workshop @ ECML-PKDD*.

Galbrun, E., & Miettinen, P. (2012b). From black and white to full color: extending redescription mining outside the boolean world. *Statistical Analysis and Data Mining*, *5*, 284–303.

Galbrun, E., & Miettinen, P. (2012c). Siren: An interactive tool for mining and visualizing geospatial redescriptions. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD 2012) (pp. 1544–1547). New York, NY, USA: ACM. doi:10.1145/2339530.2339776.

Gallo, A., Miettinen, P., & Mannila, H. (2008). Finding subgroups having several descriptions: Algorithms for redescription mining. In *Siam Conference on Data Mining (SDM 2008), Atlanta* (pp. 334–345). SIAM.

Gamberger, D., & Lavrac, N. (2002). Expert-guided subgroup discovery: Methodology and application. *J. Artif. Intell. Res. (JAIR)*, *17*, 501–527.

Gamberger, D., Mihelčić, M., & Lavrač, N. (2014). Multilayer clustering: A discovery experiment on country level trading data. In *Proceedings of the 17th International Conference on Discovery Science (DS 2014), Bled, Slovenia, October 8-10, 2014.* (pp. 87–98). Springer International Publishing. doi:10.1007/978-3-319-11812-3_8.

Grünwald, P. D. (2007). *The Minimum Description Length Principle (Adaptive Computation and Machine Learning)*. The MIT Press.

Guns, T., Nijssen, S., Zimmermann, A., & Raedt, L. D. (2011). Declarative heuristic search for pattern set mining. In M. Spiliopoulou, H. Wang, D. J. Cook, J. Pei, W. W. 0010, O. R. Zaïane, & X. Wu (Eds.), *ICDM Workshops* (pp. 1104–1111). IEEE Computer Society.

Herrera, F., Carmona, C. J., González, P., & Jesus, M. J. (2010). An overview on subgroup discovery: foundations and applications. *Knowledge and Information Systems*, *29*, 495–525. doi:10.1007/s10115-010-0356-2.

Hijmans, R. J., Cameron, S. E., Parra, J. L., Jones, P. G., & Jarvis, A. (2005). Very high resolution interpolated climate surfaces for global land areas. *International Journal of Climatology*, *25*. URL: www.worldclim.org.

Hipp, J., Güntzer, U., & Nakhaeizadeh, G. (2000). Algorithms for association rule mining — a general survey and comparison. *SIGKDD Explor. Newsl.*, *2*, 58–64. doi:10.1145/360402.360421.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Comput. Surv.*, *31*, 264–323. doi:10.1145/331499.331504.

Klösgen, W. (1996). Explora: A multipattern and multistrategy discovery assistant. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 249–271). Menlo Park, CA, USA: American Association for Artificial Intelligence.

Kocev, D., Vens, C., Struyf, J., & Džeroski, S. (2013). Tree ensembles for predicting structured outputs. *Pattern Recognition*, *46*, 817–833. doi:http://dx.doi.org/10.1016/j.patcog.2012.09.023.

Lavrač, N., Kavšek, B., Flach, P., & Todorovski, L. (2004). Subgroup discovery with cn2-sd. *J. Mach. Learn. Res.*, *5*, 153–188.

van Leeuwen, M., & Galbrun, E. (2015). Association discovery in two-view data. *IEEE Trans. Knowl. Data Eng.*, *27*, 3190–3202. doi:10.1109/TKDE.2015.2453159.

Leman, D., Feelders, A., & Knobbe, A. (2008). Exceptional model mining. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases - Part II* (ECML-PKDD 2008) (pp. 1–16). Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-540-87481-2_1.

Michalski, R. S. (1980). Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts. *Journal of Policy Analysis and Information Systems*, *4*, 219–244.

Mihelčić, M., Džeroski, S., Lavrač, N., & Šmuc, T. (2015a). Redescription mining with multi-label predictive clustering trees. In *Proceedings of the 4h workshop on New Frontiers in Mining Complex Patterns* (NFMCP '15) (pp. 86–97). Porto, Portugal.

Mihelčić, M., Džeroski, S., Lavrač, N., & Šmuc, T. (2015b). Redescription mining with multi-target predictive clustering trees. In *New Frontiers in Mining Complex Patterns - 4th International Workshop, NFMCP 2015, Held in Conjunction with ECML-PKDD 2015, Porto, Portugal, September 7, 2015, Revised Selected Papers* (pp. 125–143). doi:10.1007/978-3-319-39315-5_9.

Mihelčić, M., & Šmuc, T. (2016). Interset: Interactive redescription set exploration. In *Proceedings of the 19th International Conference on Discovery Science (DS 2016), In Press*. Bari, Italy.

Mitchell-Jones, A. (1999). *The Atlas of European Mammals*. Poyser natural history. T & AD Poyser. URL: www.european-mammals.org.

Novak, P. K., Lavrač, N., & Webb, G. I. (2009). Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *J. Mach. Learn. Res.*, *10*, 377–403.

Padmanabhan, B., & Tuzhilin, A. (1998). A belief-driven method for discovering unexpected patterns. In *Proceedings of the 4th International conference on Knowledge Discovery and Data Mining (KDD 1998)* (pp. 94–100). AAAI Press.

Parida, L., & Ramakrishnan, N. (2005). Redescription mining: Structure theory and algorithms. In *Proceedings of the 20th National Conference on Artificial Intelligence and the 17th Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA* (pp. 837–844).

Pasquier, N., Bastide, Y., Taouil, R., & Lakhal, L. (1999). Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th International Conference on Database Theory* (ICDT 1999) (pp. 398–416). London, UK, UK: Springer-Verlag.

Piccart, B. (2012). *Algorithms for Multi-Target Learning (Algoritmes voor het leren van multi-target modellen)*. Ph.D. thesis Katholieke Universiteit Leuven Belgium.

Ramakrishnan, N., Kumar, D., Mishra, B., Potts, M., & Helm, R. F. (2004). Turning CARTwheels: An alternating algorithm for mining redescriptions. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD 2004) (pp. 266–275). New York, NY, USA: ACM. doi:10.1145/1014052.1014083.

Tishby, N., Pereira, F., & Bialek, W. (1999). The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing* (pp. 368–377).

Tuzhilin, A. (1995). On subjective measures of interestingness in knowledge discovery. In *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining, (KDD 1995), Quebec, Canada* (pp. 275–281). AAAI Press.

UNCTAD (2014). Unctad database. URL: http://unctad.org/en/Pages/Statistics.aspx accessed October 2013.

Wang, H., Nie, F., & Huang, H. (2013). Multi-view clustering and feature learning via structured sparsity. In *Proceedings of the 30th International Conference on Machine Learning, (ICML 2013), Atlanta, GA, USA, 16-21 June 2013* (pp. 352–360).

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*, 236–244.

Winterfeldt, D., & Fischer, G. W. (1975). Multi-attribute utility theory: Models and assessment procedures. In D. Wendt, & C. Vlek (Eds.), *Utility, Probability, and Human Decision Making* (pp. 47–85). Dordrecht: Springer Netherlands. doi:10.1007/978-94-010-1834-0_3.

WorldBank (2014). World bank database. URL: http://data.worldbank.org/. accessed October 2013.

Wrobel, S. (1997). An algorithm for multi-relational discovery of subgroups. In J. Komorowski, & J. Zytkow (Eds.), *Principles of Data Mining and Knowledge Discovery* (pp. 78–87). Berlin / Heidelberg: Springer volume 1263 of *Lecture Notes in Computer Science*. doi:10.1007/3-540-63223-9_108.

Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, *2*, 165–193. doi:10.1007/s40745-015-0040-1.

Zaki, M. J. (2000). Scalable algorithms for association mining. *IEEE Trans. on Knowl. and Data Eng.*, *12*, 372–390. doi:10.1109/69.846291.

Zaki, M. J., & Ramakrishnan, N. (2005). Reasoning about sets using redescription mining. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (KDD 2005) (pp. 364–373). New York, NY, USA: ACM. doi:10.1145/1081870.1081912.

Zhang, M., & He, C. (2010). Survey on association rules mining algorithms. In Q. Luo (Ed.), *Advancing Computing, Communication, Control and Management* (pp. 111–118). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-05173-9_15.

Zinchenko, T. (2014). *Redescription Mining Over non-Binary Data Sets Using Decision Trees*. Master's thesis Universität des Saarlandes Saarbrücken Germany.

Chapter 6

Exploring Redescription Sets

Visualization, analysis and exploration of patterns is a very important task, which follows the task of pattern and pattern set mining. The main goal of techniques, tools and visualizations developed for this task is to allow performing different analyses, creating various summaries and obtaining knowledge that would be much harder to obtain by manual exploration of the constructed set of patterns.

Tools allowing exploration and analysis of patterns can be stand-alone tools, which are used after the patterns have been obtained, or can be integrated within the mining process, forming a basis for interactive mining tools. Interactive mining tools allow the user to actively guide the mining process towards a desired set of patterns. Such approaches require more time and focus from the user than using fully automated techniques, however it can greatly reduce the amount of generated patterns due to a very focused search. Despite the advantages of interactive mining tools, users need to be very careful with their usage. Incorrect use can significantly limit the amount of discovered knowledge, and in extreme cases constrain it to only those hypotheses expected by the user [108].

In the continuation, we provide a short survey of techniques for visualization and exploration of patterns in various data mining fields.

6.1 Tools for Interactive Exploration and Visualization of Patterns

Work presented in [108]–[110] discusses the problem of interactive data mining and pattern selection. Boley et al. [109] and van Leeuwen [110] argue that constructing tools that can fully interact with the domain expert during pattern mining, provide means for obtaining patterns interesting to the expert and allows her to incorporate domain-specific knowledge into the mining process. Since such exploration is customised and allows mining very specific patterns, it alleviates the problem of the majority of fully automated approaches which produce very large sets of patterns. Miettinen [108] encourages careful use of interactive pattern mining tools to avoid obtaining only these patterns expected by the user.

MIME [111] is a framework for interactive visual pattern mining. It allows interactive mining of itemsets and association rules. Mining is performed after the selection of evaluation measures and appropriate miners. After selecting one itemset, the best extensions are listed, which allows for creating new itemsets that can be extended further. Zaki and Phoophakdee [112] created a tool for exploring, mining and visualizing minimal association rules called MIRAGE. It uses lattice-based visualisation and exploration of minimal association rules. The association rule visualization system for exploratory data analysis [113] uses scatter plot to visualize rules from the association rule set. The rules can be explored on the individual level by modifying the rules and observing comparative bar charts prior and post attribute addition or deletion. A user-driven and quality-oriented visualization for mining association rules [114] embeds association rules to a 3D landscape. It allows the user to select rule subsets, navigate among different subsets and to filter them on several interestingness measures by using sliders. Multi-level spatial association rules mined by the tool ARES [115] are visualised by using graphs. The R programming package called the *arules* [116] allows creating and implementing transaction databases. It also offers basic algorithms for finding, analysing and visualising association rules. The package allows association rules to be visualised via scatter plots, two-key plots, 3-D matrices, grouped matrices, and in a graph form. Rule to item association rule 3-D visualization [117] allows representing several association rules and their confidence. Andrienko and Andrienko [118] developed a map-based visualization for interactive exploration of spatial data. Rojas et al. [119] created a tool that allows mining, exploring and visualization of association rules. The tool uses data table, pie chart, dot plot and parallel coordinates plot to visualize entities described by the mined association rules. It also utilizes a visualization based on the Self Organizing Map (SOM) [120] to display the spatial distribution of entities contained in the support set of the selected association rules.

Apolo [121] and TourViz [122] are two tools that enable exploration of large networks. The Apolo tool is designed to allow sensemaking (understanding) larger graphs of nodes, representing some objects. The user can arrange nodes in the network and arrange one or more nodes into distinct groups. Based on this grouping, the Apolo tool computes the relevance of other nodes (using Belief Propagation algorithm [123]), contained in the network, to the defined groups. It also allows automated placement of nodes with high relevance to the predefined groups. The TourViz tool enables displaying the connectivity graph of a set of nodes and displaying their close neighbourhood. The user can select any subset of nodes and place them in previously or newly defined groups. The tool allows observing connections between specified groups.

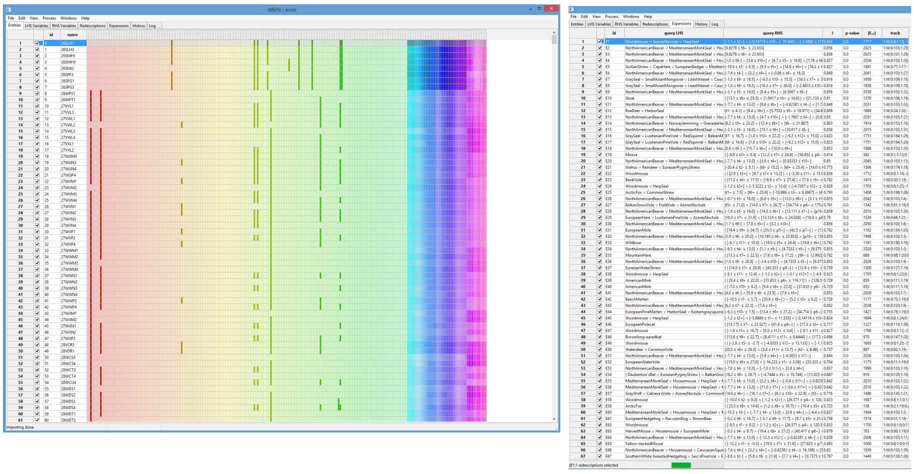
6.2 Interactive Redescription Set Exploration with Siren

The Siren tool, developed by Galbrun and Miettinen [15], [124] is an interactive redescription mining environment allowing the creation, exploration and analysis of redescriptions. It has several built-in redescription mining algorithms including implementations of the ReReMi [13], Layered trees, Split trees [14], [38] and the CARTwheels algorithm [3]. The tool visualizes the input data by showing the entity-attribute heatmap (see Figure 6.1a). Darker colors in the columns containing numerical attributes denote higher numerical values. Different categorical values of some attribute have a different shade of color which enables visually distinguishing between different categories.

Siren is designed as interactive and any time miner, however it emphasises exploration and analyses of individual redescriptions (not using the information about statistical properties of the redescription set to enhance the exploration). Redescription mining process runs and immediately displays results satisfying user-defined criteria as they are produced (see Figure 6.1b). The table displays redescription queries, Jaccard index, p-value and a support size.

Redescriptions of interest can be selected and individually analysed in more detail. They can also be extended by manually adding attributes or modifying their values (see Figure 6.2). A generated redescription can also be expanded by allowing the tool to improve the accuracy by adding new attributes to the redescription queries. Redundant redescriptions can be filtered based on redescription support.

The parallel coordinates plot (Figure 6.2a), allows for visualizing values of entities con-



a) Data visualization

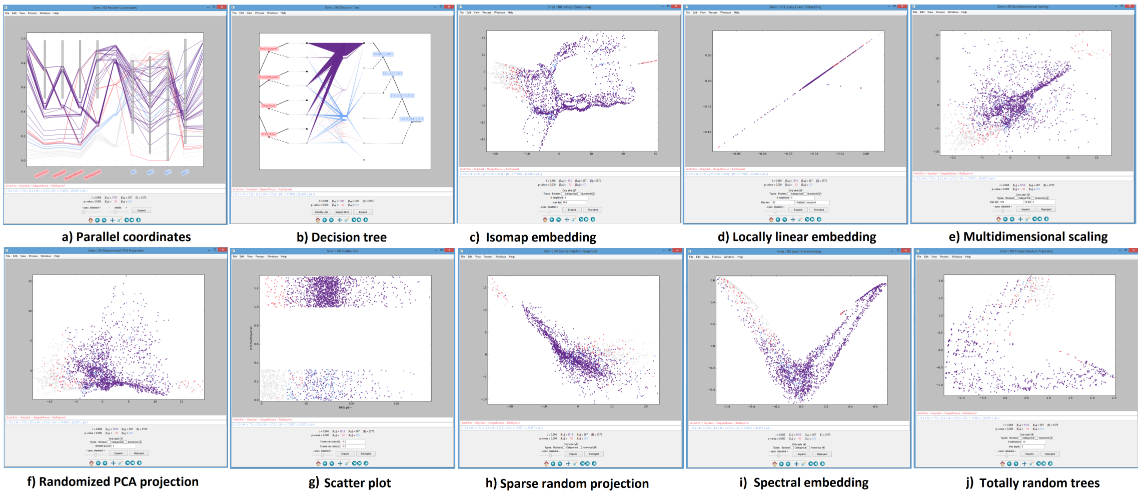
b) Process of mining redescriptions

Figure 6.1: Data visualization and redescription creation window of the Siren tool.

tained in the redescription support set, those not contained in the redescription support set but described by some redescription query and entities not described by either redescription query. The visualization is useful for observing potential regularities of entity values for some attribute contained in redescription support. The decision tree visualization (Figure 6.2b) shows the interactions between different queries contained in the redescription (especially useful to understand redescriptions created by some approach based on decision-trees). The tool also contains several entity visualization techniques based on different types of projections and embeddings (Figure 6.2c-j). The isomap and locally linear embedding (Figure 6.2c, d) are a non-linear dimensionality reduction techniques. They use different information about neighbourhood of points to embed them in lower dimensional space. Isomap uses multi dimensional scaling whereas locally linear embedding solves the eigenvector problem to find the embedding vectors. Multi dimensional scaling (Figure 6.2e) is a form of non-linear embedding that aims to preserve the distances between points, obtained in higher dimensional space, in a projected (lower-dimensional) space. Randomized PCA projection (Figure 6.2f) uses randomized version of singular value decomposition to obtain eignevectors. Top k obtained eigenvectors allow performing dimensionality reduction. The entity scatter plot (Figure 6.2g) allows comparing values of described entities based on two different attributes, which enables correlation analysis. Sparse random projection (Figure 6.2h) reduces the dimensionality of a space using a sparse random matrix. Spectral embedding (Figure 6.2i) uses affinity matrix (precomputed or computed by using some predefined method) to create a graph whose Laplacian is decomposed and the resulting (predefined) number of eigenvectors is used to perform dimensionality reduction. Totally random trees method (Figure 6.2j) constructs a forest of random trees and encodes each entity based on its occurrence in different leafs of these trees. Such, sparse dataset can be embedded in lower dimensional space using some dimensionality reduction technique, for instance singular value decomposition.

If geographical locations are described by redescriptions, the tool is able to represent the locations described by redescription queries on a map[1].

[1] More details can be seen on the Siren's home page: `http://siren.gforge.inria.fr/main/intro.html`



a) Parallel coordinates
b) Decision tree
c) Isomap embedding
d) Locally linear embedding
e) Multidimensional scaling
f) Randomized PCA projection
g) Scatter plot
h) Sparse random projection
i) Spectral embedding
j) Totally random trees

Figure 6.2: Different visualizations aimed at analysing individual redescriptions.

6.3 Interactive Redescription Set Exploration with InterSet

The InterSet tool[2] [21] uses individual information about redescriptions (redescription queries, different redescription evaluation measures, redescription support sets and the entity value distribution in these sets etc.) and information about redescription set (redescription evaluation measures using information about the structure of a redescription set, statistical analyses based on redescriptions contained in a redescription set etc.) to allow different modes of exploration and provide additional information about the observed redescription set.

The tool allows three modes of redescription set exploration:

- *Entity-based redescription set exploration*: this mode of exploration focuses on grouping entities based on similarity of their occurrence in support sets of different redescriptions. Groups of entities are arranged in a hexagonal map so that more similar groups are located closer together. Selecting one group of entities allows exploring all or different subsets of redescriptions describing all or subsets of entities contained in a selected group. Each produced group can be segmented further producing more homogeneous clusters.

- *Attribute-based redescription set exploration*: allows for analysing pairwise attribute associations based on their co-occurrence in redescriptions contained in a redescription set. Associations between more than two attributes can be observed by applying several steps of pairwise association exploration.

- *Property-based redescription set exploration*: allows for filtering redescriptions from an input redescription set based on multiple redescription quality criteria. It also allows for analysing value distributions of different redescription quality measures for these redescriptions. Distributions can also be obtained for an arbitrary subset of redescriptions.

Each redescription can be analysed individually by observing redescription queries, individual redescription quality measures and entity value distribution for each attribute occurring in redescription queries. Entity value distribution analysis includes: a) value distribution for all entities containing non-missing values in the dataset, b) value distribution

[2] http://zel.irb.hr/interset/

for all entities containing non-missing values in a numeric interval, c) value distribution for all entities contained in a redescription support set, d) value distribution for all entities that are described by at least one clause containing the given attribute, e) value distribution for all entities contained in the redescription support set described by a particular clause. To enable all modes of individual redescription analysis, the redescription queries need to be transformed to the Disjunctive Normal Form [27].

6.4 Related Publication

Details of interactive redescription set exploration and the capabilities of the InterSet tool are described in the following publication (included in this Chapter):

M. Mihelčić and T. Šmuc, "InterSet: Interactive redescription set exploration," in *Proceedings of Discovery Science: 19th International Conference, DS 2016, Bari, Italy, October 19–21, 2016*, T. Calders, M. Ceci, and D. Malerba, Eds. Cham: Springer International Publishing, 2016, pp. 35–50.

The author contributions are as follows. Matej Mihelčić devised and structured all exploration views and implemented the InterSet tool, performed the exploration on the use case dataset, presented and performed the statistical analysis of the obtained knowledge. He wrote the majority of the manuscript text. Tomislav Šmuc initiated the idea of creating the application for visualization or exploration of redescriptions and suggested using the Self Organizing Map in the entity-based exploration. He was involved in writing and correcting the manuscript text.

InterSet: Interactive redescription set exploration

Matej Mihelčić[1,2] and Tomislav Šmuc[1]

[1] Ruđer Bošković Institute
Bijenička cesta 54, 10000 Zagreb, Croatia
{matej.mihelcic, tomislav.smuc}@irb.hr
[2] Jožef Stefan International Postgraduate School
Jamova cesta 39, 1000 Ljubljana, Slovenia

Abstract. We propose a novel approach for interactive redescription set exploration and redescription analysis realized through the tool InterSet. The tool is developed for interaction with possibly large redescription sets, produced on large datasets, and it enables better understanding of the underlying data and relations between attribute sets. New insights from redescription sets can be obtained through three different interaction modes based on: i) similarity of entity occurrence in redescription support sets, ii) attribute co-occurence in redescriptions and iii) redescription quality measures. These modes provide additional contextualization, which is a major advantage compared to current state of the art approaches that allow interactive redescription set exploration, enabling users to obtain new knowledge in the form of interesting redescription subsets which can be analysed further on the level of individual redescriptions.

Keywords: knowledge discovery, redescription mining, redescription set, interactive exploration, self organising map, heatmap, crossfilter

This is a post-peer-review, pre-copy edit version of an article published in Proceedings of the DS16 conference. The final authenticated version is available online at: 10.1007/978-3-319-46307-0_3

1 Introduction

We focus our research on redescription mining [16], a field of data mining with a specific goal of finding different descriptions (called redescriptions) of similar groups of entities. These entities are described by one or more sets of Boolean, categorical or numerical attributes called views which are usually disjoint if more than one view is used. The benefits of using redescription mining are twofold: it provides information about groups of entities and means of observing connections between attributes from one or more different attribute spaces.

1.1 Notation and definition

Although redescription mining is not limited by the number of views, all current approaches (including InterSet) work with maximally two distinct views W_1 and

W_2 with the corresponding sets of variables V_1, V_2 and the set of entities E. In this setting, a redescription R is a pair of queries $R = (q_1, q_2)$ where each query describes a set of entities by using variables from the set of variables corresponding to one view. Variables in the queries are logically connected with conjunction, negation and disjunction operators.

We present one redescription obtained on our use case dataset describing world countries by using general country information and information about their trading patterns (fully described in Section 3). The redescription $R = (q_1, q_2)$ contains two queries q_1 and q_2 defined as:

$q_1:\ 23.8 \leq$ UN_YOUTH_M $\leq\ 54.4\ \wedge\ 66.4\ \leq$ STOCKS $\leq\ 166.6$

$q_2:\ 5.0\ \leq\ E_{66}\ \leq\ 6.0\ \wedge\ 4.0\ \leq\ E_{88}\ \leq\ 5.0$

Variables (UN_YOUTH_M - percentage of unemployed male youth, STOCKS - turnover ratio of traded stocks) in q_1 and (E_{66} - the percentage of total export obtained with medicinal and pharmaceutical products , E_{88}- the percentage of total export obtained with electrical machinery, apparatus and appliances) in q_2 are connected with the conjunction (AND) operator.

A single redescription is typically characterized by three quality measures: the support, the Jaccard index and the p-value.

The support of a query q_i ($supp(q_i)$) is a set of all entities satisfying its condition. The redescription $R = (q_1, q_2)$ describes the entity if this entity is in a support of all queries forming the redescription. All entities described by a redescription compose a redescription support set ($supp(R) = supp(q_1) \cap supp(q_2)$).

The intuition behind redescription mining is that queries describing similar sets of entities provide information about the shared properties of these entities. Higher similarity among sets of entities represents higher association between the queries. Thus, it is appropriate to use Jaccard index, defined as $J(R) = \frac{|supp(q_1) \cap supp(q_2)|}{|supp(q_1) \cup supp(q_2)|}$, as a measure of redescription accuracy.

The p-value (p_{val}), also used by Galbrun and Miettinen [6], reflects statistical significance of individual redescription and is computed from the binomial distribution: $p_{val}(R) = \sum_{n=|supp(R)|}^{|E|} \binom{|E|}{n} (p_1 \cdot p_2)^n \cdot (1 - p_1 \cdot p_2)^{|E|-n}$. $|E|$ equals the number of entities in the dataset and p_1, p_2 correspond to marginal probabilities of obtaining the query q_1 and q_2. For a given redescription $R = (q_1, q_2)$, $p_{val}(R)$ represents a probability of obtaining a set of a size equal to or larger than that of $supp(R)$, by combining two random queries with marginal probabilities corresponding to the marginal probabilities of queries q_1 and q_2.

We define $attr(R)$ as a set of attributes used in redescription queries and the attribute Jaccard index of two redescriptions as: $attJ(R_1, R_2) = \frac{|attr(R_1) \cap attr(R_2)|}{|attr(R_1) \cup attr(R_2)|}$. The average attribute Jaccard index of a redescription R_i is defined as: $AvgAJ(R_i) = \frac{2 \cdot \sum_{j \neq i} attJ(R_i, R_j)}{n \cdot (n-1)}$. By analogy, the entity Jaccard index of two redescriptions is defined as $elemJ(R_1, R_2) = \frac{|supp(R_1) \cap supp(R_2)|}{|supp(R_1) \cup supp(R_2)|}$ and the average entity Jaccard index as: $AvgEJ(R_i) = \frac{2 \cdot \sum_{j \neq i} elemJ(R_i, R_j)}{n \cdot (n-1)}$. These measures provide information about the redundancy of a redescription with respect to entities and attributes.

1.2 Related work

Redescription mining is an unsupervised descriptive task, closely related to multi-view clustering [3]. Redescription mining in addition to finding interesting groups of entities also provides interpretable rules describing these groups. It is also related to association rule mining [1, 10, 22] because both approaches search for relations between attributes. The main difference is that redescription mining searches for equivalence relations whereas association rule mining finds implication relations.

The first approaches developed for redescription mining [16, 21, 15] used redescription support and Jaccard index as sole constraints to limit redescription creation. Statistical significance was later incorporated into redescription mining process by Gallo et. al. [8] to further constrain redescription creation. They proposed to compute the p-value of redescriptions from the binomial distribution. Recent approaches [6, 23, 14] incorporate information about statistical significance and mostly return smaller sets of redescriptions to the final user, though the exact number of produced redescriptions varies depending on different algorithm parameters. The goal is to make redescription queries understandable and non-redundant. These approaches are able to mine redescriptions containing Boolean, categorical and numerical attributes which extends the capabilities of previous approaches that only worked with Boolean attributes. Despite the efforts to create smaller sets of accurate and understandable redescriptions, it is still hard and time consuming to analyse all produced redescriptions and their properties. It is even harder to notice potential connections between different redescriptions, their support sets and different attributes only by observing algorithm output files.

For this reason and to allow more customizable exploration process, it is necessary to develop interactive applications that respond to user inputs and provide the required information. Zaki and Ramakrishnan [21] developed a console based application that allows limited user interventions such as finding attributes describing a given set of entities, finding entities described by a given set of attributes. It also allows placing constraints on entities and attributes, Jaccard index and redescription support to allow interaction with exploration process. Siren [7] is fully interactive redescription mining environment. It allows mining redescriptions and contains several visualizations of individual redescriptions. The parallel coordinates plot, allows visualizing values of entities from redescription support, those not contained in the support but described by some redescription query and other entities not described by either redescription query. The visualization is useful to observe potential regularities of entity values for some attribute contained in redescription support. The decision tree visualization shows interactions between different queries contained in the redescription (especially useful to understand redescriptions created by some decision tree based approach). The entity scatter plot allows comparing values of described entities based on two different attributes which enables correlation analysis. If geographical locations are described by redescriptions, the tool is able to represent the locations described by redescription queries on a map. Each generated redescription can be expanded by allowing the tool to improve the accuracy by adding new attributes to the redescription queries. Redundant redescriptions can be filtered based on redescription support.

The Siren tool offers many visualizations aimed at analysis of individual redescriptions. Although very useful, the approach requires users to scroll through the list of redescriptions examining each individually to get some context about the described entities and used attributes. By performing such exploration it is hard to place each redescription in a bigger context (determine redescription relation with respect to described entities and attributes used in queries). Besides filtering, there are no mechanisms that allow grouping of different redescriptions based on their properties that allow exploring parts of redescription space that are of immediate interest to the user.

Several tools for visualizing and exploring association rules are related to our work. The Self Organizing Map (SOM) [11] is used by Rojas et. al. [5] to display the spatial distribution of the entities associated with the association rule on a map. The association rule visualization system for exploratory data analysis [13] uses scatter plot to visualize rules from the association rule set. The rules can be explored on the individual level by modifying the rules and observing comparative bar charts prior and post attribute addition or deletion. The MIRAGE [20] is a framework for mining, exploring and visualising minimal association rules. It uses lattice-based visualisation and exploration of minimal association rules. A user driven and quality oriented visualization for mining association rules [4] embeds association rules to 3D landscape. It allows users to select rule subsets, navigate among different subsets and to filter them on several interestingness measures by using sliders. Multi level spatial association rules mined by the tool ARES [2] are visualized by using graphs.

1.3 Contributions

We describe the InterSet (Figure 1), a tool aimed at interactive, comprehensive exploration and interpretation of redescription sets. The InterSet uses large diversity and potentially higher level of granularity in the redescription set to increase the usefulness of the exploration. The exploration can be done based on: (i) entities described in redescriptions from the redescription set through the SOM visualization (EC-View), (ii) attributes used in redescriptions to describe different entities through the heatmap visualization (AI-View) and (iii) quality measures assigned to individual redescriptions by using cross-filter on multiple redescription quality criteria (RQ-View). The proposed views allow contextualization, grouping and targeted exploration of different redescriptions.

The tool uses the intuition that the high overlap of entities described by redescription queries indicates existence of shared properties and possible associations between the used attributes. This property is used to build a SOM map that groups entities based on their membership in support sets of redescriptions contained in a redescription set. Resulting groups potentially share many common properties and are interesting for exploration. In addition, we obtain a spatial map of entities based on similarity of their shared properties across

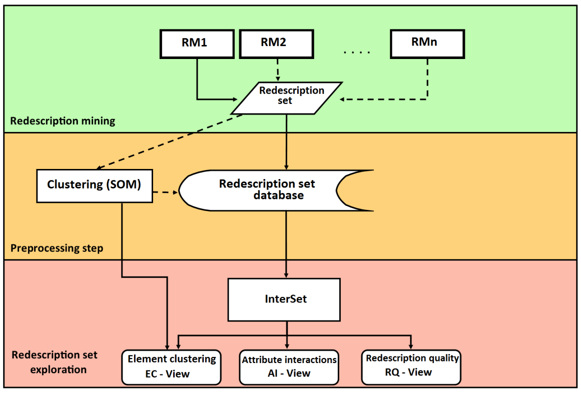


Fig. 1: Schematic description of the process that leads to interactive exploration of redescription sets: (i) Redescription set with one (mandatory) or more (optional) redescription mining algorithms must be generated, (ii) preprocessing step involves the use of Self organizing Maps clustering algorithm (optional) and database preparation (mandatory), necessary to perform (iii) interactive redescription set exploration with the InterSet tool.

both views. Attribute co-occurrence in redescription queries is used to create a heatmap of frequent cross-view pairwise attribute associations (as these are commonly observed when more than one view is used). The cross-filter visualization allows obtaining smaller set of redescriptions with some desired properties that can be explored in more detail. Redescription set obtained in each exploration view can contain a number of similar redescriptions which can be used to enhance the analysis. Comparing similar redescriptions allows understanding interactions between attributes and redescription support or detecting groups of entities with many common properties. If such level of granularity is not needed, the set can be reduced by eliminating redescriptions with large entity or attribute overlap, thus obtaining diverse and compact set with some desired properties. Except for the general insight into redescription set, the tool allows obtaining specific knowledge on the level of individual redescriptions. This includes value distribution analysis of entities contained in redescription support across all attributes contained in redescription queries and comparison with value distributions of all entities in the dataset or in some more specific groups, such as attribute numeric interval. Violin plot, used in the tool, allows visualization of irregular distribution shapes obtained when disjunction and negation operators occur in queries. The analysis allows understanding complex queries by transforming them to the Disjunctive normal form (DNF) which allows exploring parts of queries represented as clauses. These features complement and deepen the level of insight provided by the parallel coordinates plot.

2 Redescription set exploration with the tool InterSet

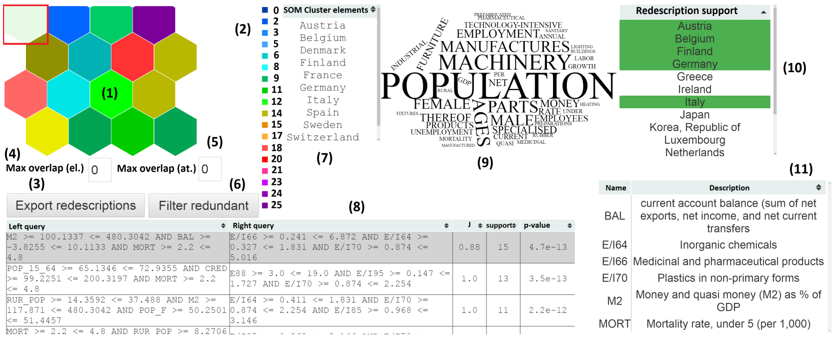
The InterSet tool (Figure 1) allows obtaining insight into some properties of the original data through different visualizations based on redescription set properties and selecting potentially interesting, non-redundant set of redescriptions suitable for detailed analysis. The selected set can be saved to a .csv file containing redescription queries and the values of corresponding quality measures.

The following subsections motivate and describe components used in redescription set exploration process for each exploration view. Tool capabilities are demonstrated on the use case data describing world countries in Section 3.

2.1 Entity based redescription set exploration

Redescription support sets usually do not have strictly hierarchical structure. Rather, they can be highly overlapping with the level of overlap depending on the underlying data, number of redescriptions in the redescription set and the algorithm used to create redescriptions. With such general structure of redescription supports, we decided to use the Self Organising Map [11] as it groups entities based on similarities and embeds similar groups closer together on a 2D visualization map. It allows representing entities from potentially large datasets in a compact form where each entity is member of only one SOM cluster.

Rather than exploiting entity similarity in the original dataset representation, as in work from Rojas et al. [5], we utilise the matrix of entity occurrence in a support set of individual redescriptions from the redescription set to obtain a map of entities sharing many cross-view properties. For a given redescription set $\mathcal{R}$ such that $|\mathcal{R}| = n$ and the original dataset containing m entities, we construct a $m \times n$ matrix A. The rows of A correspond to the entities from the original dataset, and columns correspond to the redescriptions from the redescription set. Thus, $A_{ij} = 1$ if and only if $R_j \in \mathcal{R}$ describes an entity $e_i \in E$.



SOM Cluster elements

Austria
Belgium
Denmark
Finland
France
Germany
Italy
Spain
Sweden
Switzerland

Redescription support

Austria
Belgium
Finland
Germany
Greece
Ireland
Italy
Japan
Korea, Republic of
Luxembourg
Netherlands

Left query	Right query	J	support	p-value
M2 >= 100.1337 <= 480.3042 AND BAL >= -3.8255 <= 10.1133 AND MORT >= 2.2 <= 4.8	E/I66 >= 0.241 <= 6.872 AND E/I64 >= 0.327 <= 1.831 AND E/I70 >= 0.874 <= 5.016	0.88	15	4.7e-13
POP_15_64 >= 65.1346 <= 72.9355 AND CRED >= 99.2251 <= 200.3197 AND MORT >= 2.2 <= 4.8	E88 >= 3.0 <= 19.0 AND E/I95 >= 0.147 <= 1.727 AND E/I70 >= 0.874 <= 2.254	1.0	13	3.5e-13
RUR_POP >= 14.3592 <= 37.488 AND M2 >= 117.871 <= 480.3042 AND POP_F >= 50.2501 <= 51.4457	E/I64 >= 0.411 <= 1.831 AND E/I70 >= 0.874 <= 2.254 AND E/I85 >= 0.968 <= 3.146	1.0	11	2.2e-12
MORT >= 2.2 <= 4.8 AND RUR_POP >= 8.2706				

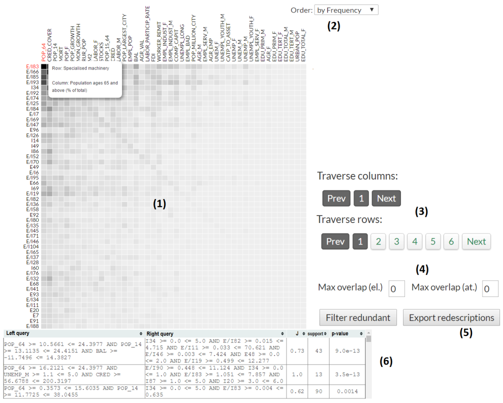
Name	Description
BAL	current account balance (sum of net exports, net income, and net current transfers
E/I64	Inorganic chemicals
E/I66	Medicinal and pharmaceutical products
E/I70	Plastics in non-primary forms
M2	Money and quasi money (M2) as % of GDP
MORT	Mortality rate, under 5 (per 1,000)

Fig. 2: The entity based interface of the InterSet tool.

The entity based redescription set exploration starts with the SOM map depicted in Figure 2, Control (1). The layout of SOM is customizable and can be easily experimented with. Each hexagon contains a distinct group of entities and the color of each hexagon reflects the number of entities contained in each group, displayed on the legend (Control (2)). The average homogeneity of a cluster, defined as the average Jaccard index between redescription support and the entities contained in the cluster, can be used as an additional cluster selection criteria. Selecting a hexagon, red square in Figure 2, provides more detailed information about it's content (Table (7)) and additional controls for in depth exploration (Table 8, Control 9). By observing information about all entities contained in the selected hexagon (Table (7)) and a Word net (9) displaying words most commonly used in attribute descriptors contained in redescription queries describing at least one entity from the SOM cluster, users can determine if it is of interest to explore all redescriptions describing at least one entity contained in the selected cluster (Table 8). Redescriptions can be analysed further on the query and the attribute level (described in Section 2.4) where Table (10) provides information about the entities described by the selected redescription (members of the SOM cluster being highlighted in green color) and Table (11) provides additional descriptions of compact attribute codes. It is possible to export (Control (3)) or filter (Controls (4), (5) and (6)) redescriptions contained in Table (8). Filtering process (described in Algorithm 1) allows obtaining a set of redescriptions with user defined maximal entity and attribute overlap.

2.2 Attribute based redescription set exploration

Research in many scientific fields such as biology, pharmacy and medicine requires discovering relevant associations between variables. Such associations can be explored with the InterSet by observing frequently co-occurring attributes in redescription queries (Figure 3).



Order: by Frequency ▼

Max overlap (el.) 0 Max overlap (at.) 0

Left query	Right query	J	support	p-value
POP_64 >= 10.5661 <= 24.3977 AND POP_14 >= 13.1135 <= 24.4151 AND BAL >= -11.7496 <= 14.3827	I34 >= 0.0 <= 5.0 AND E/I82 >= 0.015 <= 4.715 AND E/I11 >= 0.033 <= 70.621 AND E/I46 >= 0.003 <= 7.424 AND E48 >= 0.0 <= 2.0 AND E/I19 >= 0.499 <= 12.277	0.73	43	9.0e-13
POP_64 >= 16.2121 <= 24.3977 AND UNEMP_M >= 1.1 <= 5.0 AND CRED >= 56.6788 <= 200.3197	E/I90 >= 0.448 <= 11.124 AND I34 >= 0.0 <= 1.0 AND E/I83 >= 1.051 <= 7.857 AND I87 >= 1.0 <= 5.0 AND I20 >= 3.0 <= 6.0	1.0	13	3.5e-13
POP_64 >= 0.3573 <= 15.6035 AND POP_14 >= 11.7725 <= 38.0455	I34 >= 0.0 <= 5.0 AND E/I83 >= 0.004 <= 0.635	0.62	90	0.0014

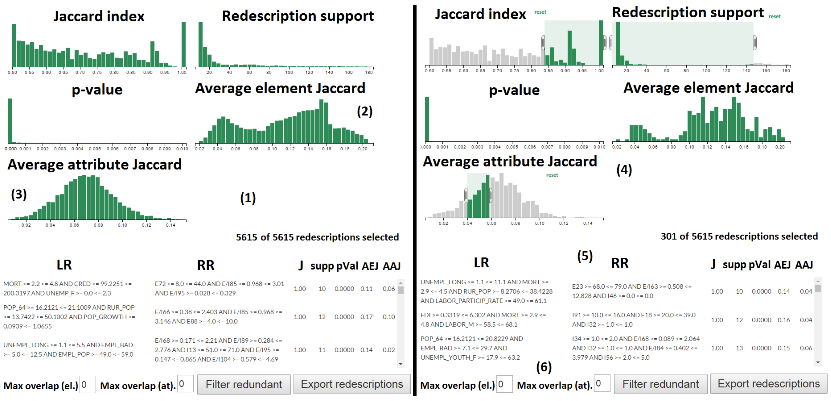
Fig. 3: The attribute based interface of the InterSet tool.

The SOM based representation can be applied to attributes used in redescription queries similarly as it was applied to entities. The main advantage of using the SOM is that it reveals interactions of more than two different variables. However, it is not possible to distinguish between views in such visualization and it is hard to explore associations between the neighbouring groups of attributes. The heatmap visualisation enables exploring interactions between all cross-view attribute pairs and arranging rows and columns based on different criteria which is the main reason for our choice. The focus is on cross-view relations since these are usually interesting when exploring similarities based on different contexts, though the tool can also be used to show all co-occurrence frequencies.

The heatmap (Control (1) in Figure 3) is a starting point of the attribute based redescription set exploration. It is represented as a $k \times s$ matrix where rows represent k attributes from the first view and columns s attributes from the second view. Three initial row-column layouts can be chosen with Control (2): 1) Ordered by name, 2) Ordered by frequency and 3) Ordered by co-occurrence. When ordered by name (useful for domain experts), the rows and columns are sorted by the attribute code, Ordered by frequency layout arranges rows and columns of the heatmap to place frequently occurring attributes in redescription queries closer to the top left corner of the heatmap. Ordered by co-occurrence layout arranges rows and columns so that it sorts the heatmap diagonal in descending order by attribute co-occurrence frequency. This layout allows finding potentially larger groups of highly connected attributes if co-occurring in redescription queries. The heatmap is adopted to be used with large number of attributes by loading smaller submatrices of the potentially large cross-view attribute matrix, whose rows are all attributes from the first view and columns all attributes from the second view. The attributes are sorted in descending order by frequency when loaded into heatmap so that the (row,column) page combination $(1, 1)$ (Control (3)) contains most frequently occurring attributes from both views. The user can scroll on two dimensions, visualizing parts of cross-view attribute space which can be explored further. The gray color denotes the co-occurrence level of the attribute pair and the table (Control (6)) equivalent to that from Figure 2 lists all redescriptions from the redescription set containing the selected attribute pair in their queries. Analysis of selected redescription is described in Section 2.4 while redescription filtering (Control (4)), (Algorithm 1) and redescription export (Control (5)) work as described in Sections 2.1 and 2.3. Combination of various attribute pair arrangements with redescription exploration and filtering allows better understanding of the attribute interactions.

2.3 Property based redescription set exploration

The last redescription set exploration view provided in the InterSet tool is based on redescription properties (quality measures). It enables users to filter the original redescription set by using one or more user-defined criteria which results in smaller, more interesting set, that is easier to explore. The exploration view uses sliders as in [4], with the important addition of a crossfilter (Control 1 in Figure 4), which allows instantaneous display of distribution of the filtered set (Control 4) for all measures and corresponding redescriptions (Table 5). Sliding through the values of one or more different criteria (Control 4), allows observing changes in distribution of other criteria which provides information about the underlying data. The visualization can be efficiently used with redescription sets containing large number of redescriptions which are much harder to represent with some other visualization techniques such as parallel coordinate plots.



5615 of 5615 redescriptions selected

LR	RR	J	supp	pVal	AEJ	AAJ
MORT >= 2.2 <= 4.8 AND CRED >= 99.2251 <= 200.3197 AND UNEMP_F >= 0.0 <= 2.3	E72 >= 8.0 <= 44.0 AND E/I85 >= 0.968 <= 3.01 AND E/I95 >= 0.028 <= 0.329	1.00	10	0.0000	0.11	0.06
POP_64 >= 16.2121 <= 21.1009 AND RUR_POP >= 13.7422 <= 50.1002 AND POP_GROWTH >= 0.0939 <= 1.0655	E/I66 >= 0.38 <= 2.403 AND E/I85 >= 0.968 <= 3.146 AND E88 >= 4.0 <= 10.0	1.00	12	0.0000	0.17	0.10
UNEMPL_LONG >= 1.1 <= 5.5 AND EMPL_BAD >= 5.0 <= 12.5 AND EMPL_POP >= 49.0 <= 59.0	E/I68 >= 0.171 <= 2.21 AND E/I89 >= 0.284 <= 2.776 AND I13 >= 51.0 <= 71.0 AND E/I95 >= 0.147 <= 0.865 AND E/I104 >= 0.579 <= 4.69	1.00	11	0.0000	0.14	0.02

Max overlap (el.) 0 Max overlap (at). 0

301 of 5615 redescriptions selected

LR	RR	J	supp	pVal	AEJ	AAJ
UNEMPL_LONG >= 1.1 <= 11.1 AND MORT >= 2.9 <= 4.5 AND RUR_POP >= 8.2706 <= 38.4228 AND LABOR_PARTICIP_RATE >= 49.0 <= 61.1	E23 >= 68.0 <= 79.0 AND E/I63 >= 0.508 <= 12.828 AND I46 >= 0.0 <= 0.0	1.00	10	0.0000	0.14	0.04
FDI >= 0.3319 <= 6.302 AND MORT >= 2.9 <= 4.8 AND LABOR_M >= 58.5 <= 68.1	I91 >= 10.0 <= 16.0 AND E18 >= 20.0 <= 39.0 AND I32 >= 1.0 <= 1.0	1.00	12	0.0000	0.16	0.04
POP_64 >= 16.2121 <= 20.8229 AND EMPL_BAD >= 7.1 <= 29.7 AND UNEMPL_YOUTH_F >= 17.9 <= 63.2	I34 >= 1.0 <= 2.0 AND E/I68 >= 0.089 <= 2.064 AND I32 >= 1.0 <= 1.0 AND E/I84 >= 0.402 <= 3.979 AND I56 >= 2.0 <= 5.0	1.00	13	0.0000	0.15	0.06

Max overlap (el.) 0 Max overlap (at). 0

Fig. 4: The InterSet interface based on redescription properties. Initial configuration is shown in the left and the filtering step in the right part of the Figure.

We use several redescription quality criteria currently used in the literature and allow adding new ones, such as different interestingness or unexpectedness criteria, to be used in the filtering process. The crossfilter (Figure 4, Control 1), consists of histograms showing value distribution for each criterion used in the exploration process. Besides previously defined standard redescription quality measures, we have computed two additional measures: the average entity Jaccard index (Control (2)) and the average attribute Jaccard index (Control (3)) presented in Section 1.1. These criteria can be used to extract redescriptions describing (in)frequently described sets of entities or that containing (in)frequent combination of attributes depending on the crossfilter setting. The selection provides no guarantees on entity or attribute overlap between pairs of redescriptions in the newly constructed set. However, filtering (described in Algorithm 1) reduces this overlap to user defined level (Control 6).

Algorithm 1 The filtering algorithm

```
Input: Redescription set $\mathcal{R}$, max entity overlap $\varepsilon_{el}$, max attribute overlap $\varepsilon_{at}$
Output: Filtered redescription set $\mathcal{R}'$
procedure FILTER
    $criteria \leftarrow ((J, desc), (attJ, asc), (supp, desc), (elemJ, asc))$
    $\mathcal{R} \leftarrow \text{sort}(\mathcal{R},\ criteria)$
    for $i = 0;\ i < |\mathcal{R}| - 1;\ i++$ do
        for $j = i + 1;\ j < |\mathcal{R}|;\ j++$ do
            if $elemJ(\mathcal{R}[i], \mathcal{R}[j]) \geq \varepsilon_{el}$ OR $attJ(\mathcal{R}[i], \mathcal{R}[j] \geq \varepsilon_{at})$ then
                $\mathcal{R} \leftarrow \mathcal{R}.\text{delete}(\mathcal{R}[j])$
    return $\mathcal{R}$
```

Criteria array from line 2 in Algorithm 1 contains pairs of redescription quality criteria and sorting direction: desc - descending, asc - ascending. Redescription with preferred values of quality criteria is used to eliminate all redescriptions with unacceptably high entity or attribute Jaccard with the selected redescription.

2.4 Analysing individual redescriptions

For detailed redescription analysis, the tool requires redescription queries to be transformed in the Disjunctive normal form (DNF). This decomposed form allows analysing distribution of a subset of redescription support described by each clause in this representation. Since the general goal is to produce short, understandable queries and every formula in Propositional logic can be transformed to an equivalent DNF [12], it is a reasonable and mostly feasible requirement aimed at increasing understandability. Depending on the query complexity, the analysis contain i) three comparative violin plots if the DNF representation doesn't contain disjunction operators (explanation 1, 2, 3 in Figure 5), ii) four plots if the DNF representation contains disjunction operators (explanation 1, 2, 3, 5 in Figure 5) and iii) five plots if an attribute occurs in more than one clause in the DNF representation of a query (explanation 1, 2, 3, 4, 5 in Figure 5).

Compared to parallel coordinates plot in Siren, our approach allows analysing value distributions and decomposed queries. Major benefit is it's invariance to dataset or redescription support size whereas parallel coordinates plot tend to have increasing number of possibly overlapping lines making analysis difficult. If Boolean or categorical values are used, violin plots are replaced with piecharts.

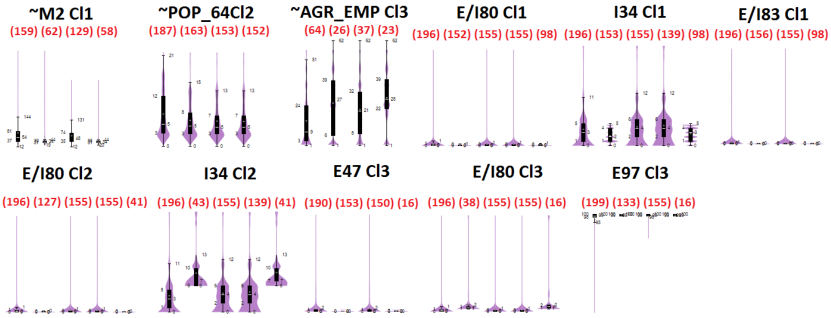


Fig. 5: Comparative violin plots showing entity value distribution for all attributes occurring in the selected redescription. The violin plots show entity distribution in order: 1) entity value distribution for the selected attribute for all entities containing non-missing values in the dataset, 2) entity distribution for all entities containing non-missing values in a numeric interval, defined in redescription query, for this attribute, 3) entity value distribution for all entities contained in redescription support set for a given attribute, 4) entity value distribution of all entities contained in redescription support set that are described by at least one clause containing the attribute under investigation, 5) entity value distribution of all entities contained in the redescription support set described by a particular clause which contains the analysed variable.

Redescription analysis process is demonstrated on redescription $R' = (q_1', q_2')$: q_1' : $\neg\,(44.1 \leq M_2 \leq 198.5 \wedge 15.5 \leq \mathrm{POP}_{64} \leq 20.8 \wedge 2.0 \leq \mathrm{AGR\_EMP} \leq 20.5)$ q_2' : $(0.0 \leq E/I_{80} \leq 0.6\ \wedge\ 0.0 \leq I_{34} \leq 5.0\ \wedge\ 0.0 \leq E/I_{83} \leq 0.6) \vee$ $(0.0 \leq E/I_{80} \leq 0.3\ \wedge\ 6.0 \leq I_{34} \leq 24.0)\ \vee\ (0.0 \leq E_{47} \leq 1.0\ \wedge\ 0.7 \leq E/I_{80} \leq 28.8\ \wedge\ 99.0 \leq E_{97} \leq 100.0)$. It describes 155 countries with $J(R') = 0.88$ and $p_{val}(R') = 0.0023$. The analysis of its queries is demonstrated in Figure 5.

Query q_1' is transformed to the equivalent formula in DNF: $\neg\,(44.1 \leq M_2 \leq 198.5)\ \vee\ \neg\,(\ 15.5 \leq \mathrm{POP}_{64} \leq 20.8)\ \vee\ \neg\,(2.0 \leq \mathrm{AGR\_EMP} \leq 20.5)$. Using negations splits the value distribution of entities in two parts. For example: $\neg\,(\ 15.5 \leq \mathrm{POP}_{64} \leq 20.8)$ is equivalent to $\mathrm{POP}_{64} < 15.5 \vee \mathrm{POP}_{64} > 20.8$. Query q_2' contains three disjunctions, two of which contain attribute I_{34} (denoted as Cl1 and Cl2). Value distributions show that first two clauses describe orthogonal entities with respect to the values of this attribute. It also shows that clause 1 constitutes a backbone of the query describing the largest amount of entities.

3 Exploring redescriptions obtained on the Country data

The tool's capabilities are demonstrated on dataset describing 199 world countries [17, 19, 9] by using general country descriptors (demographic descriptors, unemployment, etc.) as one view and country trading patterns (with the values of percentage of export (E) and import (I) that a given commodity forms in total country export or import and the information on the ratio of these values (E/I)) as the other view. All attributes in the dataset contain numerical values.

We used the redescription mining algorithm, presented in [14], to create a set containing 5448 different redescriptions where some of them, by design, had high level of similarity. For all $R \in \mathcal{R}$, $J(R) \geq 0.5$, $p_{val}(R) \leq 0.01$ and $supp(R) \geq 10$.

The SOM needs to be precomputed and the obtained cluster membership represents the input to the tool[3]. In the experiments, we used the *R* package *kohonen* [18] to create the SOM with the 4 × 4 layout. To train it, we used 1000 iterations with the learning rate linearly declining from 0.05 to 0.01.

3.1 Redescription set analysis

The results and analysis presented in this Section are obtained with the tool InterSet, available at `www.zel.irb.hr/interset`.

The largest group of countries contained in the SOM map contains 28 countries and the smallest group only 3 different countries. The number of redescriptions describing a particular cluster ranges between 612 and 3002. We explore a SOM cluster with the highest average homogeneity (0.34). This cluster, emphasized in Figure 2 is described with 2737 different redescriptions and contains 11 Western European countries. Portugal and a part of the Eastern, South - Eastern and Central European countries form a separate cluster with high homogeneity (> 0.2) which is located in the neighbouring cluster. The only other cluster with

[3] `zel.irb.hr/interset`

homogeneity higher than 0.2 contains 28 different countries mostly located in Africa and Asia. Majority of countries from the clusters presented in [9] (Table 4) are grouped together in one of our SOM clusters which tend to contain larger groups of countries spatially ordered by similarity of their shared properties.

The table containing redescriptions from the selected SOM cluster was sorted in descending order by Jaccard index and in ascending order by redescription support to search for accurate redescriptions with possibly homogeneous support. We selected the third result ($R_{ex} = (q_1'', q_2'')$) for in depth analysis:
$q_1'': -6.9 \leq \text{MON\_GR} \leq 6.6 \wedge 17.1 \leq \text{POP}_{64} \leq 21.1 \wedge 41.9 \leq \text{STOC} \leq 166.6$
$q_2'': 4.0 \leq E_{24} \leq 26.0 \wedge 3.0 \leq I_{95} \leq 5.0 \wedge 1.1 \leq E/I_{85} \leq 3.2$. This redescription describes 10 countries with $J(R_{ex}) = 1.0$ and $p_{val}(R_{ex}) = 6.3 \cdot 10^{-12}$. It describes Austria, Denmark, Finland, France, Germany, Italy, Portugal, Spain, Sweden and United Kingdom. Portugal, not a member of selected cluster, is contained in the neighbouring cluster (below). We compare value distribution of countries described by a redescription to a value distribution of all countries in the dataset with respect to attributes used in its queries. The described set of countries tend to have higher values in POP_{64} (percentage of population aged 65+). The difference is significant, as computed with the Mann-Whitney U test, with the p-value of $9.05 \cdot 10^{-7}$. These countries tend to have smaller values in MON_GR (money and quasi money growth - $p = 6.7 \cdot 10^{-5}$), higher values in STOC (stock trade - $p = 0.0002$), E_{24} (labour-intensive and resource-intensive manufactures - $p = 0.011$), I_{95} (articles of apparel and clothing accessories - $p = 1.54 \cdot 10^{-5}$) and E/I_{85} (industrial machinery and parts - $p = 2.3 \cdot 10^{-6}$). Countries from the selected SOM cluster, with the exception of Switzerland, tend to have higher values of E/I_{69} (plastics in primary form - $p = 0.0003$), E/I_{83} (specialised machinery - $p = 1.2 \cdot 10^{-6}$) and tend to have smaller values in AGR_M (percentage of mail employees working in agriculture - $p = 0.0024$). A part of discoveries related to export of technology, manufactures and population parameters match those reported in [9], here we present additional observations, which are a very small subset of information obtainable by the tool.

Attribute associations are, due to finding equivalence relations, an important and distinguishing feature of redescription mining. Attributes are ordered by co-occurrence which reveals high co-occurrence between attributes in the top left corner of the heatmap (Figure 3). We explore correlations between all pairs of attributes contained in the top - left 5 × 5 submatrix. Since the Kolmogorov-Smirnov test shows that entity values for many attributes contained in this submatrix are not normally distributed, we compute correlation values (presented in Table 1) by using Spearman's rho and Kendall's tau correlation coefficients.
Results in Table 1 show that associations between view 1 attributes: mortality rate under 5 (MORT), private credit bureau coverage (CRED_COVER), percentage of population aged 0 to 14 (POP_{14}), percentage of population aged 65+ (POP_{64}) and percentage of female population (POP_F) contain statistically significant correlations with the view 2 attributes: I_{34} (cereals and cereal products), E/I_{66} (medicinal and pharmaceutical products), E/I_{83} (specialized machinery), E/I_{85} (other industrial machinery and part) and E/I_{93} (furniture

Table 1: Spearman's ρ and Kendall's τ correlation coefficient of a selected 5×5 cross-view attribute associations. For each attribute pair, a upper bound of p-values for both correlation coefficients is displayed below the correlation values.

Sp. ρ / Ken. τ	I_{34}	$\mathrm{E/I}_{66}$	$\mathrm{E/I}_{83}$	$\mathrm{E/I}_{85}$	$\mathrm{E/I}_{93}$
MORT	$0.62/0.46$ $< 10^{-15}$	$-0.66/-0.48$ $< 10^{-15}$	$-0.65/-0.47$ $< 10^{-15}$	$-0.63/-0.46$ $< 10^{-15}$	$-0.43/-0.29$ $< 10^{-8}$
CRED_COVER	$-0.52/-0.41$ $< 10^{-11}$	$0.56/0.45$ $< 10^{-15}$	$0.43/0.32$ $< 10^{-8}$	$0.48/0.35$ $< 10^{-10}$	$0.48/0.36$ $< 10^{-10}$
POP_{14}	$0.69/0.52$ $< 10^{-15}$	$-0.66/-0.46$ $< 10^{-15}$	$-0.61/-0.42$ $< 10^{-15}$	$-0.64/-0.45$ $< 10^{-15}$	$-0.48/-0.33$ $< 10^{-10}$
POP_{64}	$-0.64/-0.48$ $< 10^{-15}$	$0.68/0.48$ $< 10^{-15}$	$0.62/0.44$ $< 10^{-15}$	$0.66/0.47$ $< 10^{-15}$	$0.49/0.33$ $< 10^{-10}$
POP_F	$-0.36/-0.25$ $< 10^{-5}$	$0.28/0.18$ $< 10^{-3}$	$0.35/0.23$ $< 10^{-5}$	$0.41/0.27$ $< 10^{-7}$	$0.37/0.26$ $< 10^{-6}$

and parts thereof). Many of this correlations might be caused by differences in country development. A number of developed countries have a high exports of all mentioned groups of commodities whereas developing countries either completely rely on the imports of this products or produce it for their own market. Since these two groups of countries differ in the population characteristics, it is possible for the correlation patterns as shown in Table 1 to emerge.

The third view is used to locate highly accurate redescriptions describing subsets of countries that are: a) different than a majority of other redescriptions with respect to described entities and attributes used in redescription queries ($AvgAJ \leq 0.05$, $AvgEJ \leq 0.1$, $JS > 0.94$), b) similar to larger number of other redescriptions with respect to described entities and attributes used in redescription queries ($AvgAJ \geq 0.1$, $AvgEJ \geq 0.15$, $JS \geq 0.94$). After obtaining each set with the cross-filter, we removed redundant redescriptions sharing more than 20% entities and 20% attributes. First experiment revealed 11 different redescriptions from which we present several interesting discoveries. The analysis revealed that 9 different African countries and a neighbouring Asian country Yemen have a smaller median of percentage of population aged 15 to 64 compared to the median of all countries, these countries also have high export contribution of textile fibres and their wastes in their total export. A subset of countries located in different SOM clusters show high export to import contribution ratio of iron, steel and chemical products. Very heterogeneous group of countries from various continents that share higher export contribution to import contribution ratio of prefabricated buildings, sanitary, heating and lighting fixtures was also discovered. Finally, a group of countries, largely comprised of eastern and south eastern European countries share a large import contribution of precious stones and non-monetary gold to their total import. Strict non-redundancy requirements left only one redescription in the second experiment describing 15 world countries that share many different socio-demographic and trading properties. The described group shows many characteristics of highly developed countries: large percentage of population older than 64, smaller percentage of rural population, higher money and quasi money (as percentage of GDP). Part of these countries have larger ratio of export to import contribution ratio of medicinal and pharmaceutical products, rubber manufactures, and a part of described countries has a large E/I ratio for other industrial machinery and parts.

Finding examples as presented in this section would be very time consuming with Siren because extensive redescription list exploration is required. Support for reasoning at the level of groups of entities, attribute associations or selecting groups of redescriptions with specific properties is not available in Siren.

4 Conclusions and future work

We have presented a tool that allows exploring potentially large redescription set obtained by one or more redescription mining approaches. It provides analytics mechanisms aimed at understanding individual redescriptions and uses the redescription set to obtain information about the underlying data - revealing connections and interactions between different entities and attributes. Potentially overlapping redescriptions are used as a tool to enhance the visualizations and allow high granularity exploration. Entity and property based exploration supports arbitrary number of data views while the attribute based view is easily extended by performing pairwise view exploration.

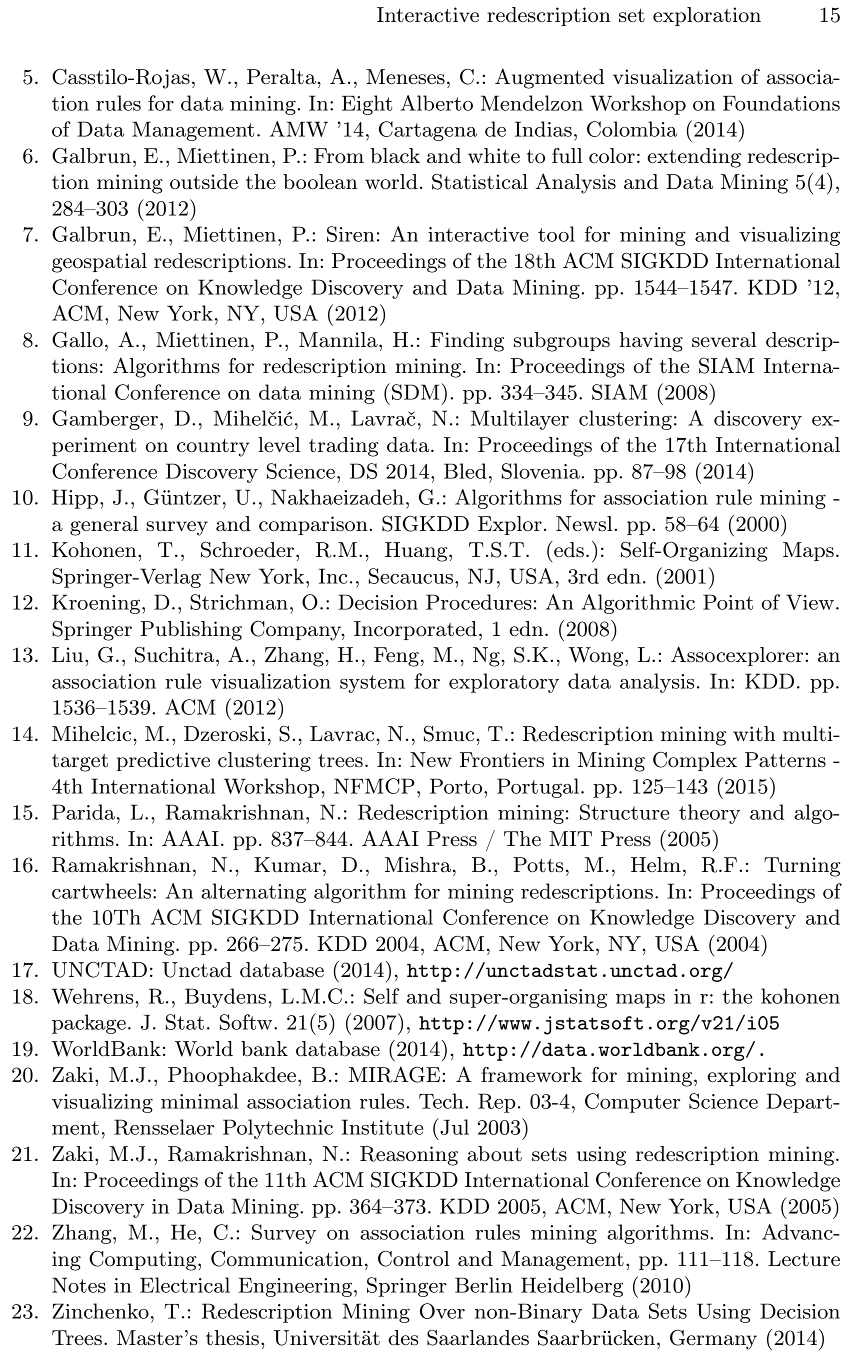
In future work we plan to increase exploration abilities of the tool by enabling different interactions between exploration views which is needed when faced with potentially large sets of redescriptions. On the technical side, we will aim to incorporate the SOM map in exploration process thus removing the need to train it separately and calling external scripts to feed the data into the database.

Acknowledgement

The authors acknowledge the European Commission's support through the MAESTRA project (Gr. no. 612944), the MULTIPLEX project (Gr.no. 317532) and support of the Croatian Science Foundation (Pr. no. 9623: Machine Learning Algorithms for Insightful Analysis of Complex Data Structures).

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Advances in Knowledge Discovery and Data Mining, pp. 307–328. American Association for Artificial Intelligence (1996)
2. Appice, A., Buono, P.: Analyzing multi-level spatial association rules through a graph-based visualization. In: IEA/AIE. Lecture Notes in Computer Science, vol. 3533, pp. 448–458. Springer (2005)
3. Bickel, S., Scheffer, T.: Multi-view clustering. In: Proceedings of the Fourth IEEE International Conference on Data Mining. pp. 19–26. ICDM '04, IEEE Computer Society, Washington, DC, USA (2004)
4. Blanchard, J., Guillet, F., Briand, H.: A user-driven and quality-oriented visualization for mining association rules. In: Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM), Melbourne, Florida, USA. pp. 493–496 (2003)



5. Casstilo-Rojas, W., Peralta, A., Meneses, C.: Augmented visualization of association rules for data mining. In: Eight Alberto Mendelzon Workshop on Foundations of Data Management. AMW '14, Cartagena de Indias, Colombia (2014)
6. Galbrun, E., Miettinen, P.: From black and white to full color: extending redescription mining outside the boolean world. Statistical Analysis and Data Mining 5(4), 284–303 (2012)
7. Galbrun, E., Miettinen, P.: Siren: An interactive tool for mining and visualizing geospatial redescriptions. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1544–1547. KDD '12, ACM, New York, NY, USA (2012)
8. Gallo, A., Miettinen, P., Mannila, H.: Finding subgroups having several descriptions: Algorithms for redescription mining. In: Proceedings of the SIAM International Conference on data mining (SDM). pp. 334–345. SIAM (2008)
9. Gamberger, D., Mihelčić, M., Lavrač, N.: Multilayer clustering: A discovery experiment on country level trading data. In: Proceedings of the 17th International Conference Discovery Science, DS 2014, Bled, Slovenia. pp. 87–98 (2014)
10. Hipp, J., Güntzer, U., Nakhaeizadeh, G.: Algorithms for association rule mining - a general survey and comparison. SIGKDD Explor. Newsl. pp. 58–64 (2000)
11. Kohonen, T., Schroeder, R.M., Huang, T.S.T. (eds.): Self-Organizing Maps. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd edn. (2001)
12. Kroening, D., Strichman, O.: Decision Procedures: An Algorithmic Point of View. Springer Publishing Company, Incorporated, 1 edn. (2008)
13. Liu, G., Suchitra, A., Zhang, H., Feng, M., Ng, S.K., Wong, L.: Assocexplorer: an association rule visualization system for exploratory data analysis. In: KDD. pp. 1536–1539. ACM (2012)
14. Mihelcic, M., Dzeroski, S., Lavrac, N., Smuc, T.: Redescription mining with multi-target predictive clustering trees. In: New Frontiers in Mining Complex Patterns - 4th International Workshop, NFMCP, Porto, Portugal. pp. 125–143 (2015)
15. Parida, L., Ramakrishnan, N.: Redescription mining: Structure theory and algorithms. In: AAAI. pp. 837–844. AAAI Press / The MIT Press (2005)
16. Ramakrishnan, N., Kumar, D., Mishra, B., Potts, M., Helm, R.F.: Turning cartwheels: An alternating algorithm for mining redescriptions. In: Proceedings of the 10Th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 266–275. KDD 2004, ACM, New York, NY, USA (2004)
17. UNCTAD: Unctad database (2014), `http://unctadstat.unctad.org/`
18. Wehrens, R., Buydens, L.M.C.: Self and super-organising maps in r: the kohonen package. J. Stat. Softw. 21(5) (2007), `http://www.jstatsoft.org/v21/i05`
19. WorldBank: World bank database (2014), `http://data.worldbank.org/.`
20. Zaki, M.J., Phoophakdee, B.: MIRAGE: A framework for mining, exploring and visualizing minimal association rules. Tech. Rep. 03-4, Computer Science Department, Rensselaer Polytechnic Institute (Jul 2003)
21. Zaki, M.J., Ramakrishnan, N.: Reasoning about sets using redescription mining. In: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. pp. 364–373. KDD 2005, ACM, New York, USA (2005)
22. Zhang, M., He, C.: Survey on association rules mining algorithms. In: Advancing Computing, Communication, Control and Management, pp. 111–118. Lecture Notes in Electrical Engineering, Springer Berlin Heidelberg (2010)
23. Zinchenko, T.: Redescription Mining Over non-Binary Data Sets Using Decision Trees. Master's thesis, Universität des Saarlandes Saarbrücken, Germany (2014)

Chapter 7

Evaluation

In this chapter we present a summary of theoretical and empirical evaluation of CLUS-RM and the redescription set optimization procedures.

7.1 Theoretical Evaluation

This section presents the complexity analysis of the CLUS-RM algorithm and its extensions. It was shown in [19] that the overall time complexity of CLUS-RM that uses *redescription set optimization by redescription exchange* technique to construct a redescription set is $O(z\cdot(|V_1|+|V_2|)\cdot|E|^2+z^2\cdot|E|)$, given a random enough hashing function is used. z denotes the number of rules produced by transforming a PCT of a depth d to rules, V_1 and V_2 denote the set of attributes contained in two views and E denotes a set of entities. The worst time complexity given inadequate hashing function is $O(z^2\cdot|E|^2+z\cdot(|V_1|+|V_2|)\cdot|E|^2)$. The redescription set optimization procedure used in this work is efficient with respect to time and space complexity. The procedure stores $|\mathcal{R}_{red}|$ redescriptions and has a time complexity of $O(|E|+|\mathcal{R}_{red}|)$.

It was shown in [18] that augmenting the mining process with a random forest of PCTs does not increase the time complexity of the CLUS-RM algorithm.

Our work presented in [20] demonstrates that the generalized redescription set construction procedure (using redescription set optimization by redescription extraction) has time complexity $O(|\mathcal{R}|\cdot|\mathcal{R}_{red}|\cdot|E|)$, where $\mathcal{R}$ denotes a set of all produced redescriptions and $\mathcal{R}_{red}$ a constructed set of redescriptions containing a user-defined number of redescriptions (when sufficiently random hashing function is used). Since the maximal number of produced redescriptions is $O(z^2)$, the overall time complexity can be written as $O(z^2\cdot|\mathcal{R}_{red}|\cdot|E|)$. It can be seen from the time complexity that it is prohibitively expensive to use this procedure when the desired output set size is very large (in the order of the size of produced redescriptions). The overall time complexity, in such cases, increases to $O(z^4\cdot|E|)$. When the desired output redescription set size is relatively small (which is the case in regular use-case scenarios), the time complexity of this procedure is $O(z^2\cdot|E|)$. If an inadequate hashing function is used, the time complexity increases to $O(z^2\cdot|\mathcal{R}_{red}|\cdot|E|^2)$. This procedure stores $O(z^2)$ redescriptions.

Time complexity comparison performed in [20] shows that CLUS-RM has equal time complexity to other tree-based approaches when the number of entities increases. Attribute size has a larger effect on computation time of greedy redescription mining algorithms than on tree-based algorithms. However, the number of entities has a larger effect on the computation time of tree-based algorithms compared to greedy algorithms.

7.2 Empirical Evaluation

Empirical evaluation performed in [19] shows that performing a larger number of iterations in the CLUS-RM algorithm improves redescription set properties (with respect to predefined redescription quality measures). Comparative performance results with the ReReMi algorithm show that CLUS-RM outperforms ReReMi on one dataset containing numerical attributes with missing values when the redescription accuracy is measured with query non-missing Jaccard index. However, it is worth noting that the ReReMi algorithm does not optimize the query non-missing Jaccard index, which may have reduced its performance. Several redescription examples produced by CLUS-RM and ReReMi algorithm are provided which enable observing the differences in a query structure due to different techniques of query construction.

Evaluation performed in [18] demonstrates that using the Random Forest of PCTs for redescription construction increases the number of produced redescriptions, their accuracy and diversity. This is reflected in the results of algorithm comparison (significantly increasing algorithm performance on a sparse dataset containing Boolean attributes).

Results described in [20] show that using conjunctive refinement significantly increases redescription accuracy. As a result, a larger number of redescriptions satisfy the minimal Jaccard index threshold, which increases the diversity of redescriptions that can be used in the construction of the output redescription set. Further experiments demonstrated that changing user-defined preference on different redescription quality criteria significantly influences the structure and properties of the output redescription set. Effects of using the variability index and changing different parameters (such as minimal redescription accuracy and support size) on the properties of the output redescription set was also empirically evaluated. These results reveal trade-offs between different redescription evaluation criteria in the output sets produced by our framework. Forcing larger redescription accuracy decreases the diversity and can reduce the total entity coverage, while increasing the redescription support set size increases entity coverage but also increases entity redundancy.

Algorithm comparative empirical evaluation (as performed in [19]) was largely extended in [18] and [20] where CLUS-RM was compared to ReReMi and two tree-based approaches: the Split trees and the Layered trees algorithms. The evaluation was performed on three different datasets, including two different accuracy measures (the pessimistic and the query non-missing Jaccard index) in the presence of missing values. Comparative evaluation performed in both works shows competitive performance of our approach to other state-of-the-art approaches. An important result is also the complementarity of redescriptions produced by CLUS-RM compared to other approaches. The complementarity of redescriptions produced by CLUS-RM was observed in the produced redescription sets which contain a high number of redescriptions constructed with only conjunction and literal level negation operators (significantly higher than other related approaches).

Examples of produced redescriptions, outlined in [18], show that CLUS-RM produced some very similar redescriptions to the ReReMi algorithm. However, many produced redescriptions show differences in the query structure depending on the redescription mining algorithm used to create them. The results presented in [22] demonstrate that redescriptions obtained with CLUS-RM provided information about many scientifically proven facts in the domain of Alzheimer's disease and indicated some scarcely explored or largely unknown research directions. Experimental setup and the evaluations performed are described in Section 8.1.

Results and discussion of predictivity/generalizability, permuatation tests and corrections for multiple hypothesis testing are presented in Appendix B.

Chapter 8

Applications

In this chapter we present the application of redescription mining and constraint-based redescription mining in the domain of medicine. Redescription mining is used to gain knowledge about the set of subjects suffering from different levels of cognitive impairment or Alzheimer's disease (AD) by using a set of clinical attributes (consisting of various cognitive tests and neuropsychological measurements) and a set of biological attributes (consisting of neuroimaging data, biospecimen and genetic data).

8.1 Mining Redescriptions of Subjects with Different Levels of Cognitive Impairment

We used the CLUS-RM algorithm [19] (see Section 4.3) to mine redescriptions in the following manner:

1. Split redescription support size into several intervals: $[5, 10]$, $[11, 39]$, $[40, 99]$ and $[100, \lceil \frac{|E|}{2} \rceil]$.

2. For each support size interval, create a redescription set containing 100 redescriptions with the CLUS-RM algorithm.

3. For each produced redescription set perform redescription accuracy and redescription support homogeneity analysis.

4. Join all redescription sets into one set containing 400 redescriptions.

5. Perform individual redescription analysis (including tests of statistical significance of difference in attribute values for subjects contained in redescription support set compared to normal control subjects).

6. Perform attribute association analysis (including tests of statistical significance of correlation between pairs of attributes).

7. Select interesting redescriptions by observing redescription support set homogeneity with respect to the level of subjects cognitive impairment.

8. Validate the obtained knowledge with published scientific papers on the topic of dementia and Alzheimer's disease.

9. Obtain additional validation, explanation and detection of interesting redescriptions by the domain expert.

10. Perform constraint-based redescription mining with CLUS-RM (discussed in Section 4.5 and introduced in [22]) to extend the knowledge about the most interesting discovery: association of Pregnancy-associated plasma protein A (PAPP-A) with different levels of cognitive impairment and AD.

Using the procedure described above we:

- Discovered subsets of patients with significant fluctuation in levels of Angiopoietin-2 (ANG2), Apolipoprotein A-II (APOAII), Brain natriuretic peptide (BNP), Ciliary neurotrophic factor (CNTF), Total blood testosterone (TSTSTRNT), Insulin, Leptin, Macrophage migration inhibitory factor (MCRPHMIF), Pregnancy-associated plasma protein A/ pappalysin-1 (PAPP-A), Pancreatic polypeptide (PPP) and Spatial Pattern of Abnormalities for Recognition of Early AD (SPARE_AD) compared to control normal (CN) subjects.
- Discovered indicators with significant fluctuations between LMCI/AD and CN subjects. The discovered indicators are: Apolipoprotein A-II (APOAII), Apolipoprotein B (APOB), Angiopoietin-2 (ANG2), Brain natriuretic peptide (BNP), Fas ligand (FASL), Leptin, Pregnancy-associated plasma protein A/ pappalysin-1 (PAPP-A) and Pancreatic polypeptide (PPP).
- Discovered significant correlations between different indicators. The list of top five associations can be seen in [22].
- Discovered association between the level of Pregnancy-associated plasma protein A (PAPP-A) and various clinical and biological indicators related to cognitive impairment and AD [22].

Many of these findings were previously discussed in the literature, although some as the connection of level of testosterone and ciliary neurotrophic factor were debated.

The most important discovery was the detection of association between PAPP-A and the level of cognitive impairment, since the potential reasons of this association were not discussed in the literature. We provide a detailed analysis of potential connection and importance of studying PAPP-A in the context of memory impairment and AD [22].

8.2 Related Publication

Details of the CLUS-RM extensions to the constraint-based redescription mining setting and the analysis of data describing subjects with different levels of cognitive impairment are described in the following publication (included in this chapter):

M. Mihelčić, G. Šimić, M. Babić Leko, N. Lavrač, S. Džeroski, T. Šmuc, and for the Alzheimer's Disease Neuroimaging Initiative, "Using redescription mining to relate clinical and biological characteristics of cognitively impaired and alzheimer's disease patients," *PLOS ONE*, vol. 12, no. 10, pp. 1–35, 2017.

The author contributions are as follows. Matej Mihelčić designed and implemented constraint-based redescription mining extensions of the CLUS-RM algorithm. He designed the experiments, performed statistical analyses of the results, searched the literature to find scientific papers that validate redescriptions. He found the scientific papers connecting PAPP-A with AD through subjects suffering from type-2 diabetes, wrote the majority of the manuscript text and the required revisions and created all supplementary material documents. Goran Šimić, as the domain expert, validated the obtained redescriptions, extended the literature with additional related medical scientific papers and wrote parts of the manuscript text extending medical explanations and providing expert evaluations of discovered redescriptions. He also provided a research hypothesis of the potential reasons for the connection of PAPP-A with AD. Mirjana Babić Leko wrote parts of the abstract and introduction of the manuscript and participated in writing and correcting revisions. Nada Lavrač suggested using the redescription mining methodology on the data describing subjects suffering from different level of cognitive impairment or AD. She actively participated in correcting the original and revised versions of the manuscript. Sašo Džeroski enabled data access and actively participated in correcting the original and revised versions of the manuscript. Tomislav Šmuc wrote parts of the introduction, discussion and conclusion, found the scientific papers connecting PAPP-A with AD through the AD connected genes and was actively involved in correcting the original and revised versions of the manuscript.

PLOS | ONE

RESEARCH ARTICLE

Using redescription mining to relate clinical and biological characteristics of cognitively impaired and Alzheimer's disease patients

Matej Mihelčić[1,2], Goran Šimić[3], Mirjana Babić Leko[3], Nada Lavrač[2,4], Sašo Džeroski[2,4], Tomislav Šmuc[1]*, for the Alzheimer's Disease Neuroimaging Initiative¶

1 Division of Electronics, Ruđer Bošković Institute, Zagreb, Croatia, **2** Jožef Stefan International Postgraduate School, Ljubljana, Slovenia, **3** Department for Neuroscience, Croatian Institute for Brain Research, University of Zagreb Medical School, Zagreb, Croatia, **4** Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia

¶ Membership of the Alzheimer's Disease Neuroimaging Initiative is provided in the Acknowledgments.
* tomislav.smuc@irb.hr

Check for updates

OPEN ACCESS

Citation: Mihelčić M, Šimić G, Babić Leko M, Lavrač N, Džeroski S, Šmuc T, et al. (2017) Using redescription mining to relate clinical and biological characteristics of cognitively impaired and Alzheimer's disease patients. PLoS ONE 12(10): e0187364. https://doi.org/10.1371/journal.pone.0187364

Editor: Ornit Chiba-Falek, Duke University, UNITED STATES

Received: September 9, 2016

Accepted: October 18, 2017

Published: October 31, 2017

Data Availability Statement: Data containing measurements of different biological and clinical attributes for different human subjects are available from the ADNI database (http://adni.loni.usc.edu/). To obtain the data, users have to register with ADNI and provide a research statement. Currently, it is not possible to for the authors to make the data publicly available because of the ADNI data use agreement (http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Data_Use_Agreement.pdf), point 3 that states: "I will not

Abstract

Based on a set of subjects and a collection of attributes obtained from the Alzheimer's Disease Neuroimaging Initiative database, we used redescription mining to find interpretable rules revealing associations between those determinants that provide insights about the Alzheimer's disease (AD). We extended the CLUS-RM redescription mining algorithm to a constraint-based redescription mining (CBRM) setting, which enables several modes of targeted exploration of specific, user-constrained associations. Redescription mining enabled finding specific constructs of clinical and biological attributes that describe many groups of subjects of different size, homogeneity and levels of cognitive impairment. We confirmed some previously known findings. However, in some instances, as with the attributes: testosterone, ciliary neurotrophic factor, brain natriuretic peptide, Fas ligand, the imaging attribute Spatial Pattern of Abnormalities for Recognition of Early AD, as well as the levels of leptin and angiopoietin-2 in plasma, we corroborated previously debatable findings or provided additional information about these variables and their association with AD pathogenesis. Moreover, applying redescription mining on ADNI data resulted with the discovery of one largely unknown attribute: the Pregnancy-Associated Protein-A (PAPP-A), which we found highly associated with cognitive impairment in AD. Statistically significant correlations ($p \leq 0.01$) were found between PAPP-A and clinical tests: Alzheimer's Disease Assessment Scale, Clinical Dementia Rating Sum of Boxes, Mini Mental State Examination, etc. The high importance of this finding lies in the fact that PAPP-A is a metalloproteinase, known to cleave insulin-like growth factor binding proteins. Since it also shares similar substrates with A Disintegrin and the Metalloproteinase family of enzymes that act as α-secretase to physiologically cleave amyloid precursor protein (APP) in the non-amyloidogenic pathway, it could be directly involved in the metabolism of APP very early during the disease course. Therefore, further studies should investigate the role of PAPP-A in the development of AD more thoroughly.

further disclose these data beyond the uses outlined in this agreement and my data use application and understand that redistribution of data in any manner is prohibited." All attribute sets used in this work are described in the appropriate electronic supplementary documents submitted with the manuscript. Short descriptions of all attributes can be seen through the Merged ADNI 1/GO/2 Packages for R located in study info section of the download data page in the ADNI database (also described in the manuscript). Viewing the data without ADNI account is not possible, however ADNI grants access to data to any interesting researcher/phd student after registration (http://adni.loni.usc.edu/data-samples/access-data/). All evidence (redescription sets) obtained as a result of this study are available as supplementary documents submitted with the manuscript.

Funding: Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from many private pharmaceutical companies and non-profit organizations. The research was partly funded by the Slovenian Research Agency, the Future and Emerging Technologies EU funded Human Brain Project (HBP, grant agreement no. 604102) – NL, SD, the European Commission funded MAESTRA project (Gr. no. 612944) - TS, SD, the Croatian Science Foundation funded projects: Machine Learning Algorithms for Insightful Analysis of Complex Data Structures (grant. no. 9623) - TS and Tau protein hyperphosphorylation, aggregation and trans-synaptic transfer in Alzheimer's disease: cerebrospinal fluid analysis and assessment of potential neuroprotective compounds (grant. no. IP-2014-09-9730) - GS.

Competing interests: The authors have declared that no competing interests exist.

Introduction

Alzheimer's Disease (AD) is an irreversible neurodegenerative disease that results in progressive deterioration of cognitive abilities and behavioural control due to synapse and neuron loss. It is the most common cause of dementia among older adults. Although available medications for treatment of mild to moderate AD (donepezil, galantamine, and rivastigmine) and severe AD (memantine) help to some level, these drugs do not modify the underlying disease process.

The Alzheimer's Disease Neuroimaging Initiative (ADNI) [1] aims to collect various imaging and biomarker data, that could be potentially useful in diagnostics and treatment of AD. The analysis of these data provides means to potentially extend our understanding of the disease, its impact on various functions of human comportment and cognitive functions, and tracking its progression.

In this work, we analysed the data obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database [1], containing clinical and biological measurements (listed in S1–S3 Files and available at http://adni.loni.usc.edu/). These measurements are taken for a set of subjects in order to test for presence of AD and the level of subjects' cognitive impairment. We divided the attributes in two main groups: clinical (clin) and biological (bio).

Clinical attributes have been obtained from numerous questionnaires and neuropsychological instruments designed to test cognition and memory with the hope of early detection of AD. These tests have been carefully designed, studied and regularly updated to increase the detection of various forms of cognitive impairment. Many such tests exist [2], but there has been no unique measure that can be used to reliably make the diagnosis [3]. Thus, combining different tests has been shown to provide more reliable results. Biological attributes have contained neuroimaging data of a number of methods to visualize brain activity, such as MRI and PET scans, along with some related and derived scores. They have also contained biospecimens: a number of blood tests and measurements, and information about the subjects' genetic markers (genetic data). These attributes have been generally considered less reliable, but are still actively investigated with the aim to aid in the early detection of AD and to help understand its complex genetic, epigenetic, and environmental landscapes.

Manual investigation of associations between attributes and analysis of their effects would require insurmountable efforts, which prompted us to use a data mining technique called redescription mining.

Work related to understanding cognitive impairment

Considerable work has been oriented towards understanding the role of biological or clinical attributes, determining correlations between different attributes and assessing their predictive power for determining the level of cognitive impairment.

Researchers have used neural imaging (MRI, PET, etc.) [4–6] to predict levels of cognitive impairment. For example, Doraiswamy et al. [7] studied PET images of subjects with cognitive decline. Donovan et al. [8] studied correlations between regional cortical thinning and worsening of apathy and hallucinations. Guo et al. [9] studied the effects of intracranial volume on association between clinical disease progression and brain atrophy or apolipoprotein E genotype. Hostage et al. [10] studied the effects of apolipoprotein E (*APOE* alleles) $\varepsilon 4$ and $\varepsilon 2$ on hippocampal volume. Other investigators have also studied the role of apolipoprotein E [11] in early mild cognitive impairment. These are just a few samples of the huge set of studies of correlations between biological, clinical attributes and the level of cognitive impairment. More extensive list can be found at http://adni.loni.usc.edu/news-publications/publications/.

Recently, Gamberger et al. used a multi-layer clustering method [12] to identify clusters of AD patients with respect to several clinical and biological attributes [3]. The same method was applied [13] to detect differences between clusters containing male and female patients. Breskvar et al. used Predictive Clustering Trees (PCTs) [14] to discover and analyse patient clusters. They focused on relations between biological features and the progression of AD by observing behavioural response of patients and their study partners (persons who are in frequent contact with the patient, study with the patient, and are able to assess the patient's functioning in daily life).

Redescription mining and related fields

In this section, we provide background information related to redescription mining and motivate its choice as a data mining technique used in our work.

The most open-ended, unsupervised data-mining technique, clustering [15–19] finds and groups similar instances based on a predefined similarity measure. It is used when underlying and possibly interesting natural grouping is unavailable, but also to reveal new groups that were previously unknown. Clustering techniques typically do not create interpretable models of data, so one has to apply other technique in order to get interpretable descriptions of induced clustering. One such approach, limited to using a single attribute set, is conceptual clustering [20, 21] that aims at finding clusters that can be described with concepts derived by using some description language.

There exists a broad group of descriptive pattern mining techniques that find and describe subsets of examples using single attribute set or view.

For example, association rule mining [22] finds associations between items (in transaction databases) or different attributes in the form of unidirectional rules. Interesting associations are typically selected based on support and confidence scores of association rules and possibly some other interestingness measures.

Subgroup discovery [23, 24] is a technique that finds queries describing groups of instances having unusual and interesting statistical properties with respect to the target variable. Contrast Set Mining [25] identifies monotone conjunctive queries that best discriminate between instances containing one target class from all other instances (e.g. subjects with diagnosis Alzheimer's Disease (AD) vs Control (CN) subjects).

In contrast to techniques operating on a single set of attributes, multi-view techniques offer advantages when the available data contains information from various sources or descriptions of different properties of instances (as is the case in this study).

Two-view data association discovery [26] aims at finding a small, non—redundant set of associations that provide insight in how two views are related. The approach can create both *bidirectional* and *unidirectional* rules as translation patterns.

Redescription mining, introduced by Ramakrishnan et al. [27], is capable of mining descriptions of subsets of data described by multiple sets of attributes. The building blocks of redescriptions are called queries (logical formulas describing a set of instances by using attributes from some particular view). Redescription queries can describe the same or very similar subset of instances with different queries, which is an important capability in the context of knowledge discovery.

Rationale for using redescription mining

Redescription mining offers advantages over related techniques and provides specific results required for our analysis. The multi-view and descriptive capabilities of redescription mining make it suitable for relating different biological attributes, many with unknown or scarcely explored role and effects on cognitive impairment, to clinical attributes designed to detect cognitive impairment and make the preliminary diagnosis.

Although a two-view data association discovery approach can be applied to this data, we aimed at discovering interesting equivalence-like associations between biological and clinical attributes on different support levels and validating them with the subjects diagnosis, that is possible with redescription mining. Two-view association discovery is also somewhat limited as it is designed to mine Boolean data and to provide small and non-redundant sets of associations (translations) between different attribute sets. In our discovery study we aim to create, potentially larger number, of understandable redescriptions that would be used as a basis for the thorough statistical analyses and the analysis performed by the domain expert.

Similar data and attributes, related to AD, have been studied before [3, 13, 14, 28]. However, this study is focussed on the analysis of the ADNI data using redescription mining, which enables using its specific advantages over other approaches to find potentially new insights and improve our understanding of the genesis of AD.

Materials and methods

This section contains descriptions of data, notation and related redescription mining approaches, CLUS-RM algorithm [29, 30] and the motivation for its use in this work. It includes description of algorithmic extensions incorporated into CLUS-RM that enable fully automated constraint-based redescription mining, where we generalize the attribute and instance level constraints introduced by Zaki and Ramakrishnan [31].

Data description

For this study, we extracted data from the ADNI database [1]. To obtain the data, we used the *Merged ADNI 1/GO/2 Packages for R* [32] located in study info section of the download data page in the database. This package contains majority of available datasets in the format of R data frames. The basis of our datasets was contained in the adnimerge data table, which contains measurement of several clinical attributes (derived by using questionnaires, observations by doctors and other tests measuring level of cognition) and biological attributes (different blood tests, genetic markers, attributes derived from brain images, volumes of different parts of the brain etc.) for 1,737 subjects. There was also a target variable—diagnosis (not used for redescription construction) containing categorical values: control normal (CN), significant memory concern (SMC), early mild cognitive impairment (EMCI), late mild cognitive impairment (LMCI) and probable AD. Values of a target variable can be considered as ordered (levels of cognitive impairment). Each subject was assigned in exactly one category and there were no missing values for this variable. By examining the subjects contained in the adnimerge data table, we have noticed two distinct groups of subjects for whom some additional distinct attributes were measured. Therefore, we created and studied three related datasets.

The distributions of patients, divided by the level of cognitive impairment, for all three datasets are provided in Table 1.

Division of attributes to clinical (clin) and biological (bio) forms two disjoint sets of attributes used as views in redescription mining. In all datasets, subjects or patients constitute the instances for the redescription mining process.

Table 2 contains full names and abbreviations for all attributes required to present our work, while Tables 3 and 4 contain corresponding basic statistical information for these attributes. Due to data normalization (especially of biological attributes), the original measuring units do not correspond to the attribute values and are not specified in the tables.

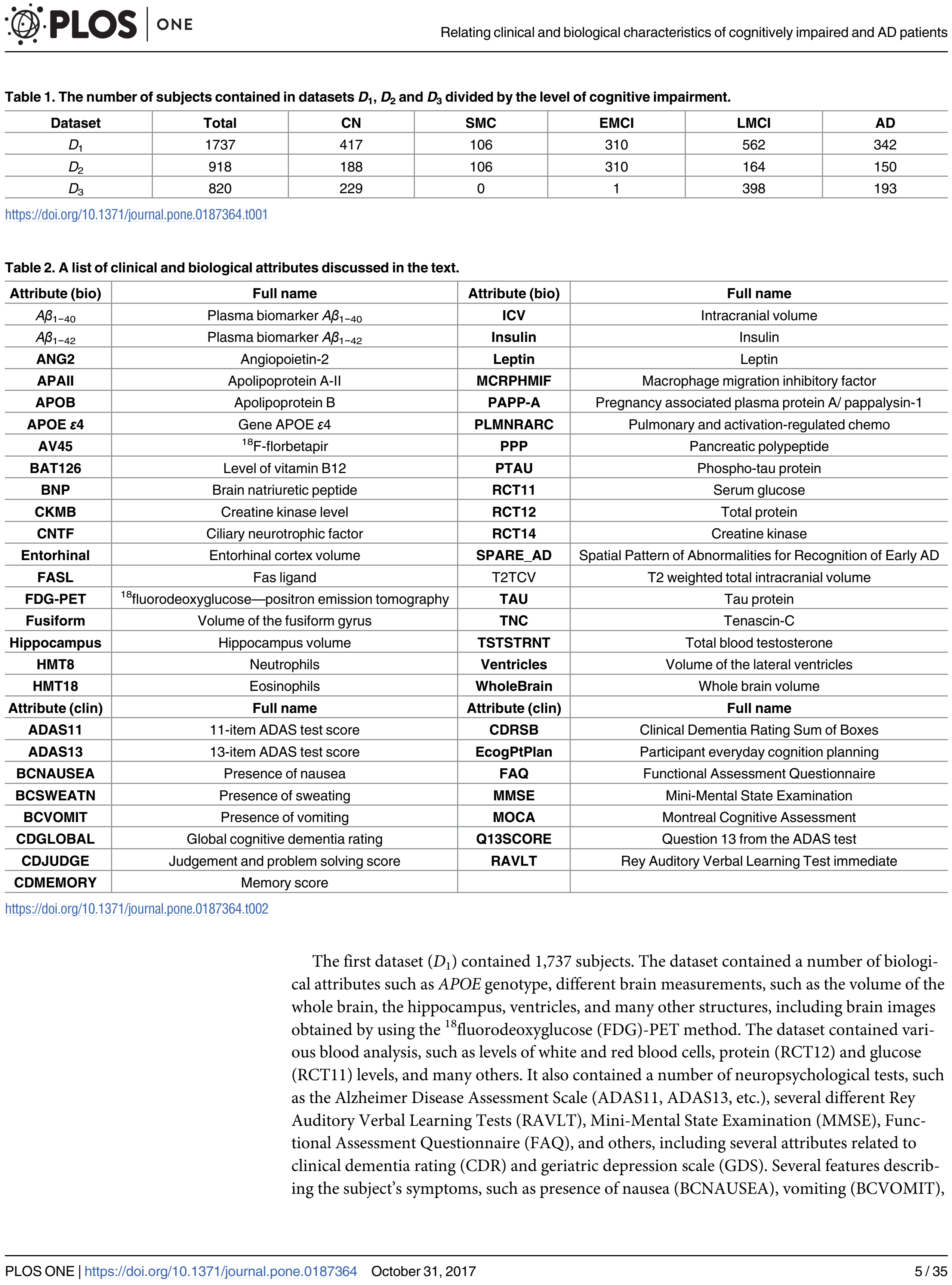


Table 1. The number of subjects contained in datasets D_1, D_2 and D_3 divided by the level of cognitive impairment.

Dataset	Total	CN	SMC	EMCI	LMCI	AD
D_1	1737	417	106	310	562	342
D_2	918	188	106	310	164	150
D_3	820	229	0	1	398	193

https://doi.org/10.1371/journal.pone.0187364.t001

Table 2. A list of clinical and biological attributes discussed in the text.

Attribute (bio)	Full name	Attribute (bio)	Full name
$A\beta_{1-40}$	Plasma biomarker $A\beta_{1-40}$	**ICV**	Intracranial volume
$A\beta_{1-42}$	Plasma biomarker $A\beta_{1-42}$	**Insulin**	Insulin
ANG2	Angiopoietin-2	**Leptin**	Leptin
APAII	Apolipoprotein A-II	**MCRPHMIF**	Macrophage migration inhibitory factor
APOB	Apolipoprotein B	**PAPP-A**	Pregnancy associated plasma protein A/ pappalysin-1
APOE ε4	Gene APOE ε4	**PLMNRARC**	Pulmonary and activation-regulated chemo
AV45	^{18}F-florbetapir	**PPP**	Pancreatic polypeptide
BAT126	Level of vitamin B12	**PTAU**	Phospho-tau protein
BNP	Brain natriuretic peptide	**RCT11**	Serum glucose
CKMB	Creatine kinase level	**RCT12**	Total protein
CNTF	Ciliary neurotrophic factor	**RCT14**	Creatine kinase
Entorhinal	Entorhinal cortex volume	**SPARE_AD**	Spatial Pattern of Abnormalities for Recognition of Early AD
FASL	Fas ligand	T2TCV	T2 weighted total intracranial volume
FDG-PET	^{18}fluorodeoxyglucose—positron emission tomography	**TAU**	Tau protein
Fusiform	Volume of the fusiform gyrus	**TNC**	Tenascin-C
Hippocampus	Hippocampus volume	**TSTSTRNT**	Total blood testosterone
HMT8	Neutrophils	**Ventricles**	Volume of the lateral ventricles
HMT18	Eosinophils	**WholeBrain**	Whole brain volume
Attribute (clin)	**Full name**	**Attribute (clin)**	**Full name**
ADAS11	11-item ADAS test score	**CDRSB**	Clinical Dementia Rating Sum of Boxes
ADAS13	13-item ADAS test score	**EcogPtPlan**	Participant everyday cognition planning
BCNAUSEA	Presence of nausea	**FAQ**	Functional Assessment Questionnaire
BCSWEATN	Presence of sweating	**MMSE**	Mini-Mental State Examination
BCVOMIT	Presence of vomiting	**MOCA**	Montreal Cognitive Assessment
CDGLOBAL	Global cognitive dementia rating	**Q13SCORE**	Question 13 from the ADAS test
CDJUDGE	Judgement and problem solving score	**RAVLT**	Rey Auditory Verbal Learning Test immediate
CDMEMORY	Memory score		

https://doi.org/10.1371/journal.pone.0187364.t002

The first dataset (D_1) contained 1,737 subjects. The dataset contained a number of biological attributes such as *APOE* genotype, different brain measurements, such as the volume of the whole brain, the hippocampus, ventricles, and many other structures, including brain images obtained by using the ^{18}fluorodeoxyglucose (FDG)-PET method. The dataset contained various blood analysis, such as levels of white and red blood cells, protein (RCT12) and glucose (RCT11) levels, and many others. It also contained a number of neuropsychological tests, such as the Alzheimer Disease Assessment Scale (ADAS11, ADAS13, etc.), several different Rey Auditory Verbal Learning Tests (RAVLT), Mini-Mental State Examination (MMSE), Functional Assessment Questionnaire (FAQ), and others, including several attributes related to clinical dementia rating (CDR) and geriatric depression scale (GDS). Several features describing the subject's symptoms, such as presence of nausea (BCNAUSEA), vomiting (BCVOMIT),

Table 3. Information about value range and percentage of missing values for biological attributes discussed in the text. Absence of an attribute from a dataset is denoted with "-" in the range and missing columns.

Attribute	D_1		D_2		D_3	
	Range	Missing	Range	Missing	Range	Missing
APOE ε4	{0, 1, 2}	1%	{0, 1, 2}	2%	{0, 1, 2}	0%
BAT126	[96, 6725]	12%	[96, 6725]	15%	[99, 3429]	8%
Entorhinal	[1426, 5896]	16%	[1438, 5896]	13%	[1426, 5731]	39%
Fusiform	[8991, 29950]	16%	[10012, 29950]	13%	[8991, 24788]	39%
Hippocampus	[2991, 10769]	14%	[2991, 10602]	10%	[3091, 10769]	19%
HMT8	[0.98, 11.64]	12%	[1.22, 10.22]	15%	[0.98, 11.64]	7%
HMT18	[0,35.8]	12%	[0, 24]	15%	[0,34.8]	7%
ICV	$[1.1, 2.1] \cdot 10^6$	1%	$[1.1, 2.1] \cdot 10^6$	2%	$[1.1, 2.1] \cdot 10^6$	0%
RCT11	[55, 413]	11%	[61, 315]	15%	[55, 413]	6%
RCT12	[5.7,9.7]	11%	[5.9,8.4]	15%	[5.7,9.7]	6%
RCT14	[18, 2658]	11%	[23, 2658]	15%	[18, 721]	6%
Ventricles	$[0.6, 1.5] \cdot 10^5$	5%	$[0.6, 1.3] \cdot 10^5$	7%	$[0.6, 1.5] \cdot 10^5$	2%
WholeBrain	$[0.7, 1.5] \cdot 10^7$	3%	$[0.8, 1.5] \cdot 10^7$	4%	$[0.7, 1.4] \cdot 10^7$	1%
AV45	[0.84, 2.03]	49%	[0.84, 2.03]	3%	-	-
FDG-PET	[3.49,8.54]	25%	[3.49,8.54]	2%	-	-
PTAU	-	-	[9.4, 173.3]	58%	-	-
$A\beta_{1-40}$	-	-	-	-	[13.0,371.8]	13%
$A\beta_{1-42}$	-	-	-	-	[4.6, 102.8]	12%
ANG2	-	-	-	-	[0.11, 1.46]	31%
APOAII	-	-	-	-	[2.35,3.18]	31%
APOB	-	-	-	-	[2.89,3.47]	31%
BNP	-	-	-	-	[1.86,4.13]	31%
CKMB	-	-	-	-	[−1.43,0.59]	31%
CNTF	-	-	-	-	[0.88,3.48]	31%
FASL	-	-	-	-	[0.85,3.62]	31%
Insulin	-	-	-	-	[−0.68, 1.43]	31%
Leptin	-	-	-	-	[−0.82, 2.0]	31%
MCRPHMIF	-	-	-	-	[−1.2,0.8]	31%
PAPP-A	-	-	-	-	[−2.34, −0.85]	31%
PLMNRARC	-	-	-	-	[1.6, 2.7]	31%
PPP	-	-	-	-	[−0.004,3.16]	31%
SPARE_AD	-	-	-	-	[−3.86, 2.79]	0%
T2TCV	-	-	-	-	[1003, 1922]	1%
TAU	-	-	-	-	[19.9,300.5]	58%
TNC	-	-	-	-	[1.9,3.5]	31%
TSTSTRNT	-	-	-	-	[−1.44, 1.52]	31%

https://doi.org/10.1371/journal.pone.0187364.t003

sweating (BCSWEATN), as well as results of various neurological examinations were also included. Information about attributes and subjects contained in D_1 are available in S1 File.

The second dataset (D_2) contained 918 subjects. In addition to features contained in the first dataset, it also contained features describing subjects' performance on Montreal Cognitive Assessment (MOCA) scale and features related to the Eastern Cooperative Oncology Group (ECOG) Scale of Performance Status. It also contained values of cerebrospinal fluid (CSF), total tau (TAU) and phospho-tau (PTAU) levels. Information about attributes and subjects contained in D_2 are available in S2 File.

Table 4. Information about value range and percentage of missing values for clinical attributes discussed in the text. Absence of an attribute from a dataset is denoted with "-" in the range and missing columns. If some dataset has equal range as D_1, this is denoted with "-||-" in the appropriate field.

Attribute	D_1		D_2		D_3	
	Range	Missing	Range	Missing	Range	Missing
ADAS11	[0,42.67]	0%	[0, 40]	0%	[0, 40]	0%
ADAS13	[0,54.67]	1%	[0, 52]	1%	[0, 52]	1%
BCNAUSEA	{0, 1}	0%	-||-	0%	-||-	0%
BCSWEATN	{0, 1}	0%	-||-	0%	-||-	0%
BCVOMIT	{0, 1}	0%	-||-	0%	-||-	0%
CDGLOBAL	{0, 0.5, . . .2}	0%	-||-	0%	{0, 0.5, 1}	0%
CDJUDGE	{0, 0.5, . . ., 3}	0%	-||-	0%	-||-	0%
CDMEMORY	{0, 0.5, . . ., 3}	0%	-||-	0%	{0, . . ., 2}	0%
CDRSB	{0, 0.5, . . ., 10}	0%	-||-	0%	{0, . . ., 9}	0%
FAQ	{0, 1, . . ., 30}	1%	{0, 1, . . ., 28}	1%	-||-	0%
MMSE	{18, 19, . . ., 30}	0%	{19, . . ., 30}	0%	-||-	0%
Q13SCORE	{0, 0.5, . . ., 10}	1%	-||-	0%	-||-	1%
RAVLT	{0, 1, . . ., 71}	0%	{1, . . ., 71}	0%	{0, . . ., 69}	0%
EcogPtPlan	-	-	[1, 4]	1%	-	-
MOCA	-	-	{4, 5, . . ., 30}	1%	-	-

https://doi.org/10.1371/journal.pone.0187364.t004

The third dataset (D_3) contained 820 subjects. It was extremely useful to study the differences and special properties of healthy subjects as compared to patients with severe stages of dementia. This dataset lacked information about ECOG Scale of Performance Status, MOCA, and information about CSF biomarkers, but it contained several additional attributes related to hormones and proteins measured. It also contained information about T2 weighted total cranial vault segmentation (T2TCV) and plasma biomarkers $A\beta_{1-40}$ and $A\beta_{1-42}$. One particularly useful imaging attribute was Spatial Pattern of Abnormalities for recognition of early AD (SPARE_AD), which was specifically constructed to help in early detection of AD. Dataset D_3 also contained the attribute PAPP-A which is analysed in more detail in this work. The AD assessment scale contained many additional attributes corresponding to different cognitive tasks, the full set of attributes being publicly available on the ADNI web page http://adni.loni.usc.edu/. Information about attributes and subjects contained in D_3 are available in S3 File.

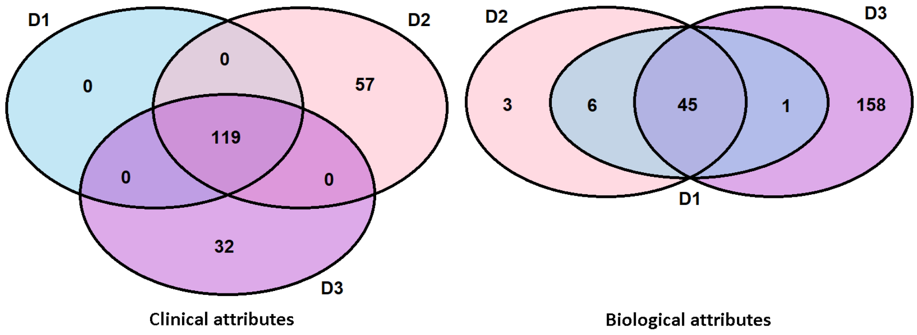


Fig 1. Relations between attributes used in constructed datasets D_1, D_2 and D_3. Left Venn diagram depicts clinical and right Venn diagram biological attributes.

https://doi.org/10.1371/journal.pone.0187364.g001

Relation between attributes used in different datasets is visible in Fig 1.

Division among subjects in the constructed datasets is as follows: $D_1 = D_2 \cup D_3$, $D_2 \cap D_3 = \{2002\}$, where 2002 denotes the roster id (RID), unique id of subject contained in the intersection.

In all analysed datasets, there were slightly more males than females. Males constitute 55% of the first, 52% of the second and 58% of the third dataset. They also constitute 57%, 53% and 61% of all subjects with some level of cognitive impairment in these datasets. Pregnancy in female subjects can alter levels of PAPP-A attribute. Although the information about the pregnancy status for female subjects analysed was not directly available in our dataset, documents describing ADNI1 exclusion criteria (which cover patients contained in our dataset D_3) [33] clearly state that female participants must be sterile or two years past childbearing potential to be included in the study group. Documents related to ADNIGO [34] and ADNI2 exclusion criteria [35] state that the participant must not be pregnant, lactating or of childbearing potential. As a result of these exclusion criteria, we can assume that the PAPP-A levels, for the studied female subjects, were not influenced by pregnancy.

Redescription mining

Redescription mining [27] works on a dataset D, containing $|D|$ instances and one set, or two disjoint sets of attributes (views, denoted as W_1 and W_2) describing these instances. A redescription (as for example $R = (q_1, q_2)$) is a pair of queries, containing one query per view. Each query is a propositional logic formula that can contain conjunction, disjunction or negation operators and is used to define conditions on values of a subset of attributes from a particular view. The subset of instances described by a query q_i, denoted $supp(q_i)$ is called the query support set. The support set of a redescription is the set of instances described by both queries that constitute this redescription: $supp(R) = supp(q_1) \cap supp(q_2)$. We also use the notation $E_{1,1}$ to denote the set of instances described by both queries, $E_{1,0}$ a set of instances described by the first query but not described by the second query, $E_{0,1}$ a set of instances described by the second query but not described by the first query, $E_{0,0}$ a set of instances that are not described by either query. $E_{?,1}$ denotes a set of instances for which it is not possible to determine if they are described by the first query, due to missing values, but are described by the second query, $E_{1,?}$ contains a set of instances described by the first query but for which it is not possible to determine if they are described by the second query, due to missing values, $E_{?,0}$ denotes a set of instances for which it is not possible to determine if they are described by the first query, due to missing values, and are not described by the second query, $E_{0,?}$ contains a set of instances not described by the first query but for which it is not possible to determine if they are described by the second query, due to missing values. The set $E_{?,?}$ contains instances for which it is not possible to determine if they are described by either query due to missing values. $attr(R)$ denotes a multiset of attributes contained in redescription queries, whereas $attrs(R)$ represents a corresponding set of attributes. $attr(D)$ denotes all attributes contained in both views of the dataset and $\mathcal{R}$ denotes a redescription set.

We evaluate the quality of mined redescriptions by using two measures [36]: i) the Jaccard index, which measures the similarity of support sets of the two redescription queries (also often called accuracy of redescription, since it measures how close two query support sets are to containing identical set of instances) and ii) statistical significance of the observed redescription, expressed through a p-value.

The Jaccard index is defined as:

$$J(R) = \frac{|supp(q_1) \cap supp(q_2)|}{|supp(q_1) \cup supp(q_2)|}$$

Assessment of the statistical significance of the redescription $R = (q_1, q_2)$ is based on an assumption that the support sets, of two queries q_1 and q_2, are selected randomly, with marginal probabilities $p_1 = \frac{|supp(q_1)|}{|D|}$ and $p_2 = \frac{|supp(q_2)|}{|D|}$ respectively. The statistical significance of redescription measures how probable it is to obtain overlap of the size $|supp(R)|$ or larger when sampling two subsets of instances from a set of size $|D|$, using sampling probabilities p_1 and p_2 respectively. The size of the intersection follows a binomial distribution and the probability we are looking for can hence be written as:

$$pV(R) = \sum_{n=|supp(R)|}^{|D|} \binom{|D|}{n} (p_1 \cdot p_2)^n \cdot (1 - p_1 \cdot p_2)^{|D|-n}$$

Example 1. Redescription $R_{ex} = (q_{clin}, q_{bio})$, discovered on dataset D_3, whose queries are defined as: q_{clin}: $0.0 \leq$ GDTOTAL $\leq 2.0 \land$ GDALIVE $= 0.0 \land$ CDMEMORY $= 0.0$ q_{bio}: $0.5 \leq$ HMT18 $\leq 16.0 \land -3.86 \leq$ SPARE_AD ≤ -0.93, provides alternative descriptions of 156 different normal control subjects. Query q_{clin} describes 204 subjects with specific value for the following clinical attributes: memory score (CDMEMORY), total score in geriatric depression scale (GDTOTAL), score on a question *Do you think its wonderful to be alive now?* (GDALIVE) while query q_{bio} describes 172 subjects having specific values for biological attributes such as percentage of Eosinophils (HMT18) and a Spatial Pattern of Abnormalities for Recognition of Early Alzheimer's disease (SPARE_AD). The set of subjects described by at least one query of redescription R_{ex} contains 220 subjects, i.e $|supp(q_{clin}) \cup supp(q_{bio})| = 220$. For 156 of 220 subjects, both queries are valid, i.e. $|supp(q_{clin}) \cap supp(q_{bio})| = 156$. This means that the Jaccard index (accuracy) for this redescription is $\frac{156}{220} = 0.709$. The redescription is statistically significant with the p-value $< 2 \cdot 10^{-17}$ (which can be computed by using the formula above). It means that it is highly unlikely to observe a redescription of support size 156 or larger given that we combine two statistically independent queries, with marginal probabilities $p_1 = \frac{204}{820} = 0.25$ and $p_2 = \frac{172}{820} = 0.21$, into a redescription R_{ex}.

Existing approaches for redescription mining. The first algorithm for redescription mining, called CARTwheels, was developed by Ramakrishnan et al. [27]. Several redescription mining algorithms have been developed since, all of which can handle Boolean attributes. From these, some algorithms [29, 30, 37, 38] work also with categorical and numerical attributes. Currently, only two redescription mining algorithms ReReMi [37] and CLUS-RM [29, 30], work with attributes containing missing values.

Redescription mining algorithms can be divided into three main categories: a) algorithms based on itemset mining, b) greedy algorithms and c) tree-based algorithms.

Itemset mining based redescription mining algorithms utilize different itemset mining methods to create itemsets, which are used to create redescriptions. Approach by Zaki and Ramakrishnan [31] and the approach by Parida and Ramakrishnan [39], use a lattice (partially ordered set) of attribute sets to find redescriptions. Approach developed by Gallo et al. [40] is based on frequent itemset mining. The field is known as Frequent Itemset Mining, because the notion of frequency (support size, the apriori principle) is central in obtaining practical algorithms.

Greedy algorithms for redescription mining work by incrementally updating queries with the goal of increasing redescription accuracy. The first algorithm developed in this category was the greedy algorithm from Gallo et al. [40]. This algorithm has been extended by Galbrun and Miettinen [37], under the name ReReMi, to work with categorical and numerical attributes.

Tree-based algorithms use decision trees [41] or Predictive Clustering trees (PCTs) [42] to create redescriptions. This category includes the first developed algorithm for redescription mining called CARTwheels, developed by Ramakrishnan et al. [27]. This algorithm works by building two decision trees per iteration (one for each view) that are joined in the leaves. Redescriptions are created by reading off the conditions along the paths from the root node of the first tree to some specified class (which constitutes one redescription query) and the paths from the root node to the matching leafs of the second tree (which constitutes the second redescription query). All created trees are of the same predefined depth, and the process iterates for a predefined number of iterations. This algorithm uses multiclass classification to guide the search between the two views. Layered trees (LayeredT) and Split trees (SplitT) algorithms developed by Zinchenko [38] use a different methodology of decision tree construction to obtain redescriptions. Instead of creating fully grown trees of predefined depth, the Layered trees algorithm creates one or more depth one trees at each algorithm step. For each leaf of the tree under construction, at some fixed iteration, the Layered trees algorithm builds a new depth one tree and appends it to the corresponding leaf of the existing tree (thus increasing its complexity and size). The algorithm allows creating more informed splits, since at a certain step of tree construction, the algorithm uses information about splits created at a corresponding level of the tree constructed on the opposite view. To construct a tree of maximal depth, the algorithm considers all nodes of the tree created on the opposite view (not just the leaves of a fully grown tree as in CARTwheels). The Split trees algorithm creates decision trees of increasing size. At each step of tree construction, the depth is increased by one and a whole new tree of larger depth is built (completely replacing the previously constructed tree) until trees of maximally allowed depth are built. This algorithm simultaneously refines classes (since it obtains finer splits with trees of larger depth) and trees (by increasing their complexity and providing more specific classes).

The CLUS-RM algorithm developed by Mihelcic et al. [29, 30] uses multi-target Predictive Clustering trees (PCTs) [42, 43], instead of decision trees to construct redescriptions. Using multi-target PCTs allows using information about all nodes (intermediate nodes as well as leaves) in the constructed PCT simultaneously to create redescriptions (which increases accuracy, diversity and number of produced redescriptions). This algorithm has been extended by Mihelcic et al. [44] to use a random forest of PCTs which further increases accuracy and diversity of produced redescriptions. The CLUS-RM is also equipped with a redescription set construction procedure called redescription set optimization [29, 30, 44]. It enables incorporating quality constraints in multi-objective optimization manner and uses all produced redescriptions to create a reduced redescription set of user-defined size. A generalized version of redescription set optimization has been presented by Mihelcic et al. [45]. In addition to its main purpose of redescription set construction, this procedure allows for use of ensembles of redescription mining algorithms, influencing the structure of produced sets through user-defined importance weights and performing computationally efficient construction of multiple redescription sets with different properties, which is beneficial for exploration [45].

Choice of methodology, redescription accuracy measure and a query language

In this section, we describe our motivation underlying the use of CLUS-RM algorithm and the extensions made to allow performing constraint-based redescription mining. In addition, we describe what reasons motivated us for the use of a redescription accuracy evaluation measure and a specific query language used to construct redescriptions.

Choice of redescription mining algorithm. To create redescriptions, we used the CLUS-RM algorithm [29, 30] based on Predictive Clustering trees (PCT) [42, 43]. PCTs allow clustering on both target and descriptive space. By using their multi-label and multi-target capability one can use multiple (or all) nodes in a given tree simultaneously to produce redescriptions. Due to the property of inductive transfer [46], multi-target classification can outperform single-target classification, which improves the overall accuracy of produced redescriptions. The CLUS-RM algorithm incorporates a redescription set optimization procedure (a novelty compared to other redescription mining approaches), which uses the large number of diverse redescriptions produced to optimize a redescription set of user-define size.

Using a large number of produced redescriptions in the optimization process increases the quality of the redescription set presented to the user. The optimization process evaluates redescriptions according to accuracy, significance and redundancy (with respect to redescription support sets and attributes contained in redescription queries).

Since our data contain missing values, we could only use the CLUS-RM or the ReReMi algorithm to find redescriptions. Given our goal of using the produced redescription sets to perform further statistical analysis, there are several reasons that motivate the use of CLUS-RM as the redescription mining algorithm in this work. CLUS-RM has the ability to produce potentially large sets of redescriptions that can be used to perform statistical analysis (e.g. of obtained associations). Multiple different redescriptions containing the same attribute pair and describing different subsets of instances reinforce the importance of frequently co-occurring attributes. CLUS-RM can constrain redescription support set size to an interval, which provides experts with a range of associations (hypotheses), from general (intervals containing larger support set size) to more specific (intervals containing smaller support set size). It can also produce redescription sets of user defined size which allows creating sets that contain equal number of members per support interval for further statistical analysis. Because of this, association statistics will not be constructed only from very general or very specific redescriptions, but from redescriptions covering a whole range of different support sizes. The experiments with CLUS-RM [30], and its extension [44], as well as the integration of the CLUS-RM into a redescription mining framework for redescription set construction [45], show that the produced redescription sets were fully competitive with other state-of-the-art solutions, and in some cases (as when only conjunctions are used in redescription query construction), the resulting redescription sets can even contain significantly more accurate and diverse redescriptions.

To obtain the results presented in this work, we required the constraint-based redescription mining capability, mostly using one attribute as constraint. However, developing a constraint-based methodology that is able to use multiple attributes (instances) as constraint was straightforward and is also presented as a part of this work. The proposed extensions include several modes of constraint-based redescription mining (CBRM) that allow exploring interactions of multiple attributes from different views with Boolean, categorical and numerical variables, extending the state-of-the-art in CBRM. Instance level constraints can be incorporated in analogous fashion.

The one-attribute CBRM capability of Siren [47] allows selecting one attribute as constraint and defining its numerical interval (for numerical attributes). The resulting redescription set is comprised of redescriptions that are obtained by extending the initial query supplied by the user. When compared to this limited CBRM capability of Siren, the CLUS-RM extension operates in a fully automated constraint-based setting (allowing multiple attributes as constraints). Also, it is not necessary to manually select numerical bounds as is currently the case in Siren. In general, performing interactive constraint-based redescription mining can demand significant effort and time from the domain expert (in addition to examination of computed redescriptions, which also needs to be done in our approach), but can potentially enable tuning the algorithm better to find information about some specific, targeted problem.

Analysis and exploration of precomputed redescription sets, based on multiple different redescription criteria, exploration of different attribute associations and groupings of instances based on a produced redescription set is also possible with the tool InterSet [48].

Choice of redescription accuracy measure. Since the data contains missing values, we used the query non-missing Jaccard index, introduced in [30], and further explained in [45] to evaluate redescriptions. The query non missing Jaccard index is defined as:

$$J_{qnm}(q_1, q_2) = \frac{|E_{1,1}|}{|E_{1,1}| + |E_{?,1}| + |E_{1,?}| + |E_{0,1}| + |E_{1,0}|}$$

Query non-missing Jaccard index evaluates instances as being a part of redescription support set only if there is enough information in the data (given the query language) to deduce that these instances satisfy the conditions of both redescription queries. The construction of this measure is guided by the principle that the query cannot contain an instance in its support set if it cannot be evaluated due to missing values. Because of this, the measure does not penalize the score with instances contained in the sets $E_{?,?}$, $E_{0,?}$, $E_{?,0}$ and rather treats them as if they were contained in the set $E_{0,0}$ but penalizes the score with instances contained in the sets $E_{?,1}$ and $E_{1,?}$ and treats them as if they are contained in sets $E_{1,0}$ and $E_{0,1}$.

Query non-missing Jaccard index has been designed to trade-off between the pessimistic and the optimistic Jaccard index [36], which are each forcing opposite extreme values and are thus leading to less realistic estimates of the true Jaccard index. Query non-missing Jaccard index is optimistic because it does not penalize the score with instances that are not described by one query and cannot be evaluated by the other query, due to missing values ($E_{?,0}$, $E_{0,?}$). On the other hand, it is pessimistic, since it penalizes the score with instances that are described by one redescription query, but cannot be evaluated by the other, due to missing values ($E_{1,?}$ and $E_{?,1}$). Redescription accuracy estimates provided by query non-missing, pessimistic and optimistic Jaccard index have already been compared experimentally in [45].

Choice of a query language. In this work, our redescriptions consist exclusively of conjunctive queries. Queries containing only conjunction operators are easier to understand and usually shorter than those containing combination of all operators. In redescriptions with queries containing only conjunction operators, every attribute used in its queries must describe all instances from redescription support set. Thus, such redescriptions discover stronger associations between attributes than redescriptions with queries containing all operators. These reasons make us believe that applying CLUS-RM with restriction to use of conjunctions to ADNI data is the right choice which may reveal useful medical hypotheses that can be further developed by the domain experts. Described query language is similar to the one used in bi-directional association rules which can, for instance, be produced by the two-view data association discovery approach, discussed in the Introduction section. In general, using negation and disjunction operators in redescription construction can increase the diversity and accuracy of produced redescriptions, but it can also make them more difficult to understand for domain experts.

CLUS-RM algorithm description

All experiments were performed with the CLUS-RM redescription mining algorithm [29, 30], presented in Algorithm 1. CLUS-RM uses PCTs [43] to find descriptions of groups of instances (i.e. subjects, as is the case in our medical study).

Algorithm 1 The CLUS-RM algorithm

```
Require: First view ($W_1$), Second view ($W_2$), maxIter, Quality constraints $\mathcal{Q}$
Ensure: A set of redescriptions $\mathcal{R}$
 1: procedure CLUS-RM
 2:   $[W_1^{(0)}, W_2^{(0)}] \leftarrow$ createInitalData($W_1$, $W_2$)
 3:   $[P_{W_1^{(0)}}, P_{W_2^{(0)}}] \leftarrow$ createInitialPCTs($W_1^{(0)}$, $W_2^{(0)}$)
 4:   $[r_{W_1^{(0)}}, r_{W_2^{(0)}}] \leftarrow$ extractRulesFromPCT($P_{W_1^{(0)}}, P_{W_2^{(0)}}$)
 5:   for Ind $\in$ {1, ..., maxIter} do
 6:     $[W_1^{(Ind)}, W_2^{(Ind)}] \leftarrow$ constructTargets($r_{W_1^{(Ind-1)}}$, $r_{W_2^{(Ind-1)}}$)
 7:     $[P_{W_1^{(Ind)}}, P_{W_2^{(Ind)}}] \leftarrow$ createPCTs($W_1^{(Ind)}, W_2^{(Ind)}$)
 8:     $[r_{W_1}^{(Ind)}, r_{W_2}^{(Ind)}] \leftarrow$ extractRulesFromPCT ($P_{W_1^{(Ind)}}, P_{W_2^{(Ind)}}$)
 9:     for ($R_{new} \in r_{W_1^{(Ind)}} \times_{\mathcal{Q}} r_{W_2^{(Ind-1)}} \cup r_{W_1^{(Ind-1)}} \times_{\mathcal{Q}} r_{W_2^{(Ind)}}$) do
10:       $\mathcal{R} \leftarrow$ addReplaceDiscard($R_{new}, \mathcal{R}$)
11:   $\mathcal{R} \leftarrow$ minimizeQueries($\mathcal{R}$)
12:   return $\mathcal{R}$
```

The presented algorithm pseudocode describes the CLUS-RM functionality in case only conjunction logical operators are used to create redescription queries. The extended version of the algorithm pseudocode for the case in which conjunction, negation and disjunction logical operators can be used in redescription query construction is described in [30] and supplementary document S18 File.

The algorithm consists of four main parts: 1) Initialization, 2) Query creation (divided in query construction 2.1 and query exploration 2.2), 3) Redescription creation and 4) Redescription set optimisation.

1) In the initialization phase (line 2 in Algorithm 1), the algorithm makes a copy of each instance from the original dataset and shuffles the attribute values for the copies. For each attribute, the algorithm selects a random instance from the dataset and copies its value for the selected attribute to the target copy (value of one instance from the original dataset can be copied multiple times). This procedure breaks correlations between attributes in the copied instances. Each instance from the original dataset is assigned a target value 1.0 and each artificially created instance a target value 0.0. It is possible to use the PCT algorithm to create initial clusters, from such dataset, by distinguishing between original instances and copies containing shuffled values (line 3 in Algorithm 1). The described procedure is repeated independently for both views contained in the dataset.

2.1) Each node in the obtained PCTs represents a cluster. These nodes are transformed to rules (line 4 in Algorithm 1) which are valid for the corresponding group of instances. More details about transforming PCTs to rules can be seen in [49].

2.2) The next step is to describe the same groups of instances, as those described by the produced rules, with the second attribute set (lines 6–8 in Algorithm 1). To do this, for each instance of the original dataset, the algorithm constructs a set of target variables containing equal number of targets as number of rules constructed using the first set of attributes (for more details see [30]). The instance has a target value 1 on position *j* if it is described by the *j*-th rule from a set of rules constructed on the first set of attributes, otherwise the value is 0. Instances for which information is missing, making it impossible to determine the membership in support set of the query are also labelled with 0. We use the multi-target classification and regression capability of PCT to construct clusters on different views containing similar instances. The procedure is repeated by creating initial rules on the second view and describing similar sets of instances by using attributes from the first view.

3) Once the algorithm obtains rules for both views, it combines them by computing the Cartesian product of two rule-sets (line 9 in Algorithm 1). Each redescription is evaluated with various user predefined constraints (such as minimal redescription accuracy, minimal support, maximal *p*-value, contained in a set of redescription quality constraints $\mathcal{Q}$), to select candidates for redescription set optimization.

4) Each redescription satisfying all user-defined redescription quality constraints is a candidate for redescription set optimization (line 10 in Algorithm 1). Satisfactory redescriptions are added to the redescription set, in the order of creation, until the maximal number of redescriptions (user-defined parameter) is reached. When this number is reached, the algorithm computes the score difference, defined in [29, 30], between the new redescription and every redescription already contained in the redescription set based on redescription score. The score of some redescription $R \in \mathcal{R}$, based on its support set and a redescription set $\mathcal{R}$, is computed as:

$$redScoreInst(R) = \frac{\sum_{i \in supp(R)}(coverInst_{\mathcal{R}\setminus R}(i))}{\sum_{i \in D} coverInst_{\mathcal{R}}(i)}$$

where $coverInst_{\mathcal{R}}(i) = |\{R \in \mathcal{R}, i \in supp(R)\}|$ denotes the number of times, the instance i is described by redescriptions from the redescription set $\mathcal{R}$. The denominator of a score *redScoreInst*(*R*) can be also written as $\sum_{R \in \mathcal{R}}|supp(R)|$. Similarly, the redescription score:

$$redScoreAttr(R) = \frac{\sum_{a \in attr(R)}(coverAttr_{\mathcal{R}\setminus R}(a))}{\sum_{a \in attr(D)} coverAttr_{\mathcal{R}}(a)}$$

is based on attributes contained in redescription queries, where $coverAttr_{\mathcal{R}}(a) = |\{R \in \mathcal{R}, a \in attr(R)\}|$ denotes the number of times attribute a is used in queries of redescriptions contained in $\mathcal{R}$. The denominator of a score *redScoreAttr*(*R*) can be also written as $\sum_{R \in \mathcal{R}}|attr(R)|$.

The score of a newly created redescription R_{new} is computed in the same way as the score for some $R \in \mathcal{R}$ but using frequencies for all redescriptions contained in the set $\mathcal{R}$ in the numerator of *redScore* and *redScoreAttr*.

The error score is computed as *errSc*(*R*) = 1.0 − *J*(*R*) and the final redescription score is computed as:

$$sc(R) = \alpha_1 \cdot errSc(R) + \alpha_2 \cdot redScoreInst(R) + \alpha_3 \cdot redScoreAt(R)$$

where $\alpha_i \in [0, 1]$, $\sum_{k=1}^{3} \alpha_i = 1$. Lower total redescription score is favourable because it implies smaller error in redescription accuracy and smaller level of instance and attribute redundancy with respect to other redescriptions from the set $\mathcal{R}$. The user—defined weights α_k regulate importance of different scores which affect the properties of the resulting redescription set. In this work, we use $\alpha_k = \frac{1}{3}$. Redescription contained in the redescription set with the highest score difference with the newly created redescription is replaced thus improving the overall redescription set quality. At each redescription exchange all frequency scores are updated.

The minimization procedure introduced in [30] and performed in line 11 of Algorithm 1 is a heuristic procedure designed to reduce the size of redescription queries by removing redundant attributes (attributes that can be removed without changing redescription accuracy). It is performed individually on each redescription of the resulting redescription set.

Constraint-based redescription mining. In this work, we extended the CLUS-RM algorithm to a constraint-based redescription mining setting. The algorithm incorporates constraints in redescription creation and one additional score in the optimization function used for redescription set creation. Necessary CBRM extensions of the CLUS-RM algorithm, when conjunction, negation and disjunction operator can be used in redescription query construction are described in supplementary document S18 File.

We present the attribute level constraints useful for gaining knowledge as demonstrated in this work. Constraints involving instances can be introduced in the analogous fashion by using redescription support set (*supp*(*R*)) instead of attribute set (*attrs*(*R*)) in formulas (1), (2) and (3).

Constraint-based redescription mining, first defined in [31], allows placing constraints on attributes that must occur in redescription queries or instances that must be contained in redescription support set. The constraints are in the form $\mathcal{C} = \{C_1, C_2, \cdots, C_n\}$, where each constraint C_i specifies a set of attributes that must occur in redescription queries or a set of instances that must be contained in redescription support set. In the original formulation, at least one constraint C_i must be satisfied by a redescription (contain all attributes or instances specified in the set) to be presented to the user. We denote this original definition as strict constrained-based redescription mining and mostly use it in our study. In practice, various relaxed versions of constrained-based redescription mining might be useful. In the continuation, we specify one existing (strict) and two newly defined (soft and suggested) modes of constraint-based redescription mining (focusing only on attribute constraints):

1. Strict constraint-based redescription mining: there must exist at least one constraint $C_i \in \mathcal{C}$ such that all defined attributes occur in redescription queries.

2. Soft constraint-based redescription mining: there must exist at least one constraint $C_i \in \mathcal{C}$ such that a part of defined attributes occurs in redescription queries. Satisfying larger portion of constraints is favoured by the redescription evaluation score.

3. Suggested constraint-based redescription mining: defined constraints are used as suggestions that increase the overall redescription score depending on the number of satisfied constraints, however high quality redescriptions not satisfying any of these constraints can also enter redescription set if their total score is high enough.

Strict constraint-based redescription mining can be used when the expert already has a hypothesis (obtained through domain knowledge and extensive experimentation) and wants to explore the specified associations in more detail. Soft constraint-based redescription mining can be used when a set of attributes of interest has been determined (by applying the combination of domain knowledge and experimentation) but it is not clear which interactions from the set should be fully explored. Thus, further study of their interactions is needed to form, refine or confirm the expert hypothesis. Suggested constraint-based redescription mining can be used when the expert, knowing the research domain (having a priori knowledge about the problem), selects a set of attributes that are known or suspected to be (currently) more interesting for exploration, though at current stage there is no immediate focus on any particular hypothesis.

To allow constraint-based redescription mining, we extend the CLUS-RM algorithm by adding a new set of constraints $\mathcal{C}$ containing the user-defined attributes of special interest and a type of CBRM used (parameter $\mathcal{T}$). Line 9 of Algorithm 1 is changed to $R_{new} \in (r_{W_1}^{(ind)})_{\{\mathcal{C},\mathcal{T}\}} \times_Q (r_{W_2}^{(ind)})_{\{\mathcal{C},\mathcal{T}\}}$. Thus, redescriptions are created only by combining those queries that satisfy predefined constraints. For each redescription R_{new}, we apply query minimization procedure before using redescription set optimization (defined in line 10 of Algorithm 1).

If query minimization procedure removes any of the key constraint attributes, defined in set $\mathcal{C}$ of CBRM, the created redescription is discarded.

In addition, CLUS-RM is extended with a new score *scConstr*, which is used in suggested constraint-based redescription mining to increase the overall score of a redescription satisfying user-defined attribute constraints. The score is defined as:

$$scConstr(R) = \frac{1}{2} \cdot max\left\{\frac{|attrs(R) \cap C_i|}{|C_i|},\ C_i \in \mathcal{C}\right\} + \frac{1}{2} \cdot \frac{|attrs(R) \cap (\cup_i C_i)|}{|attrs(R)|} \tag{1}$$

The first term in the score rewards redescriptions satisfying higher fraction of constraints from some set C_i. Due to the fact that more disjoint or partially overlapping constraint sets can be given and the fact that some redescriptions can satisfy parts of larger number of constraint sets C_i, we take the maximum score achieved among constraint sets as a quality of redescription—thus favouring compliance with larger number of constraints from a single constraint set. The second term favours redescriptions that, among the attributes contained in their queries, have larger fraction of attributes of interest to the user. Here, we reward satisfied constraints from any constraint set defined by the user.

The score used for soft constraint-based redescription mining is defined as:

$$redScoreSoft = \begin{cases} scConstr(R) & \text{if } \exists C_i \in \mathcal{C}, attrs(R) \cap C_i \neq \emptyset \\ -\infty & \text{otherwise} \end{cases} \tag{2}$$

Similarly, the score used for strict constraint-based redescription mining is defined as:

$$redScoreStrict = \begin{cases} 1 & \text{if } \exists C_i \in \mathcal{C}, attrs(R) \cap C_i = C_i \\ -\infty & \text{otherwise} \end{cases} \tag{3}$$

Higher scores denote higher level of agreement of redescriptions with the imposed constraints (redescriptions with higher score are thus preferable).

Finally, redescription score $sc(R)$ is extended to:

$$\begin{aligned} sc(R) \quad = \quad & \alpha_1 \cdot errSc(R) + \alpha_2 \cdot redScoreInst(R) + \\ & +\alpha_3 \cdot redScoreAt(R) + \alpha_4 \cdot (1 - redScoreConst(R)) \end{aligned}$$

where $\alpha_i \in [0, 1]$, $\sum_{k=1}^{4} \alpha_i = 1$ and *redScoreConst*(*R*) denotes any variant of the constraint-based score chosen by the user. Redescriptions with the score value of ∞ are not allowed to enter redescription set.

With the extension introduced above, the CLUS-RM is the only redescription mining algorithm capable of performing fully automated constraint-based redescription mining on categorical, numerical and data containing missing values with more than one attribute constraint.

Experiments and results

In this section, we present the experimental setup and some selected results obtained through the analyses of the produced redescription sets.

Experiments

Our main goal was to study clinical and biological attributes, and to find interesting relations among them. To retrieve maximum information from and to obtain deeper insight into the data, we divided redescriptions by the number of described subjects and used the diagnosis of the level of cognitive impairment to further assess the relevance and interestingness of the obtained redescriptions. For each dataset, we created four redescription sets containing redescriptions with different supports, describing [5, 10], [11, 39], [40, 99] and at least 100 subjects. The maximum support threshold was set to $\left\lceil \frac{|D_i|}{2} \right\rceil$ subjects contained in the dataset D_i, $i \in \{1, 2,3\}$. We are interested in re-describing subsets of subjects with some level of cognitive impairment and using cognitively normal subjects as a control group. Studying different biological, clinical attributes and their interactions in the context of different levels of cognitive impairment is also of high interest. Higher homogeneity of described subjects increases the amount of information obtained about different changes in biological and clinical attributes occurring as a result of different level of cognitive impairment. Developing an approach with a combined properties of redescription mining and subgroup discovery may also be interesting in this setting, but is beyond the scope of this work. Each set contains 100 redescriptions with a minimal Jaccard Index of 0.2 and a maximal p-value of 0.01. Allowed support intervals, as well as other parameter limits were found through experimentation. Redescriptions contained up to 8 attributes per query.

The same support intervals were used to create redescriptions on each dataset. This allows making easier comparisons of redescriptions and statistics of attribute co-occurrence across different datasets. Distribution analysis of redescription quality measures, in the produced redescription sets, reveals potentially interesting datasets, attributes and support intervals.

Since PAPP-A showed interesting associations with cognitive impairment in the experiments described above, we performed constraint-based redescription mining with the same algorithmic parameters but focusing redescription search on redescriptions containing pregnancy associated plasma protein A (PAPP-A) in the redescription queries. We created one redescription set containing 100 redescriptions describing at least 100 subjects.

Redescription accuracy and homogeneity analysis

We merged the four sets of redescriptions, of different supports, created on each dataset (D_1, D_2, D_3) and formed one larger redescription set (RS) per dataset, denoted $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$ (see Fig 2). For the obtained redescriptions, contained in the corresponding redescription sets ($\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$), we analysed the homogeneity of the described subsets of subjects with respect to the degree of cognitive impairment (CN, SMC, EMCI, LMCI and AD) by computing the entropy of described subject's medical diagnosis (demonstated in Fig 2).

The entropy was computed for the support set of each redescription by using the package *entropy* developed for the programming language *R*. The package allows estimating Shannon's entropy ($H = -\sum_{i=0}^{N-1} p_i \cdot log_2(p_i)$) [50] of some finite set of probabilities obtained from the observed counts (occurrence frequencies of each level of cognitive impairment in the redescription support set). In this use-case, N equals the number of different target classes occurring in the support set of a redescription. Probability p_i is computed as $p_i = \frac{|\text{target}_i \cap supp(R)|}{|supp(R)|}$, where $\text{target}_{i,\, i \in \{0, \ldots, N-1\}}$ denotes a set of entities with target label i. Characteristics of redescription sets produced with different support intervals (1., 2., 3., 4. in Fig 2), can be seen on a plot showing entropy distributions (i in Fig 2) and distributions of redescriptions' Jaccard index (ii in Fig 2).

Due to the smaller diversity in target classes (containing no SMC subjects and only 1 EMCI subject), it was easier to distinguish between different groups of subjects on dataset 3 (which is illustrated in Fig 2) than on the other two datasets. On dataset 3, we obtained many clusters of various size, homogeneous with respect to medical diagnosis, which gives us confidence that we found attribute combinations and numerical intervals useful for the analysis and understanding of cognitive impairment connected to AD.

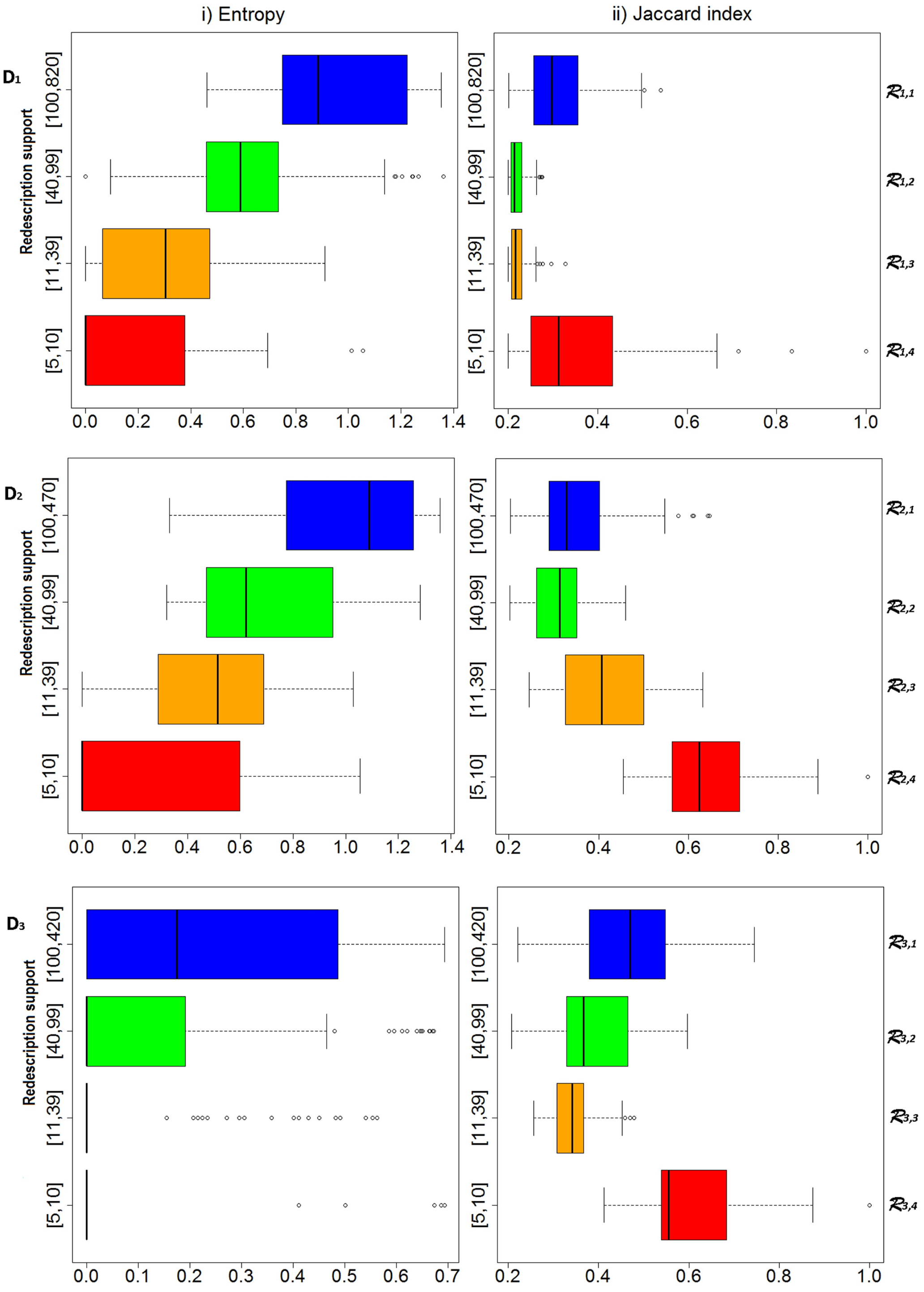

i) Entropy
ii) Jaccard index
D1
D2
D3
Redescription support
[100,820]
[40,99]
[11,39]
[5,10]
[100,470]
[100,420]
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
1.0
1.2
1.4
$\mathcal{R}_{1,1}$
$\mathcal{R}_{1,2}$
$\mathcal{R}_{1,3}$
$\mathcal{R}_{1,4}$
$\mathcal{R}_{2,1}$
$\mathcal{R}_{2,2}$
$\mathcal{R}_{2,3}$
$\mathcal{R}_{2,4}$
$\mathcal{R}_{3,1}$
$\mathcal{R}_{3,2}$
$\mathcal{R}_{3,3}$
$\mathcal{R}_{3,4}$

Fig 2. Entropy (i) and Jaccard index (ii) value distributions for the redescription sets created on each dataset (first dataset—D_1 at the top, third dataset—D_3 at the bottom). For a dataset D_i, $i \in \{1, 2,3\}$, we create four redescription sets $\mathcal{R}_{i,1} - \mathcal{R}_{i,4}$ so that the number of described subjects in each redescription (from a particular redescription set) falls in the corresponding interval shown on the y-axis (boxplots representing distributions for each interval are coloured in different color). Each redescription set $\mathcal{R}_{i,j}, i \in \{1, 2, 3\}, j \in \{1, \cdots, 4\}$ contains 100 redescriptions.

https://doi.org/10.1371/journal.pone.0187364.g002

The entropy increases with the increase of the number of described subjects, while the Jaccard index shows stronger associations in redescriptions with support in the first ($|supp(R)| \geq 100$ in Fig 2) and the last interval ($|supp(R)| \in [5, 10]$ in Fig 2). Redescriptions describing the smallest number of subjects (the last interval) use larger number of attributes with very specific numerical intervals to isolate groups of subjects that are very homogeneous with respect to the medical diagnosis and describe many different groups of subjects suffering from severe cognitive impairment (LMCI, AD). In contrast, many accurate redescriptions (in the first interval) use larger numerical intervals, thus often describing subjects with various levels of cognitive impairment. Additional reason for higher accuracy in this interval compared to the middle two intervals is the detection of highly accurate redescriptions describing subgroups of CN subjects. Missing values in the data and potential noise, occurring from the errors in measurements and data processing, negatively affect the Jaccard index.

Analyses based on examination of redescription sets

Redescription set analyses, which included: a) the examination and expert evaluation of individual redescriptions, b) the distribution analysis of level of dementia for the described subjects of these redescriptions, c) comparative analyses of attribute value distribution between different subsets of subjects (LMCI/AD vs CN or *supp*(*R*) vs CN), allowed us to find useful information related to subjects with cognitive impairment.

From the clinical attributes, we noticed that ADAS, MOCA, Geriatric Depression Scale, Rey Auditory Verbal Learning Test (especially the percent forgetting score), and Mini-Mental State Exam (MMSE) occurred frequently in queries of obtained redescriptions that describe subjects suffering from various degrees of cognitive impairment. Nevertheless, there were instances where some CN subjects fell in the identified intervals of values for these attributes. Attributes connected to Clinical Dementia Rating distinguished well between CN subjects and those with different degrees of cognitive impairment. Redescriptions mostly contained the attributes CDMEMORY, CDGLOBAL and CDR-SB (clinical dementia rating sum of boxes). From the biological attributes, we often encountered attributes connected to brain volume, hippocampus, various blood and urinary tests (attributes HMT and RCT), intracranial volume (ICV), ^{18}fluorodeoxyglucose—positron emission tomography (FDG-PET) and ^{18}F-florbetapir (AV45). These attributes have been studied before by Gamberger et al. [3, 13]. We noticed that the biological attribute SPARE_AD (Spatial Pattern of Abnormalities for Recognition of Early AD) correlated with subject's diagnosis very well and occurred frequently in redescriptions constructed on dataset 3 that contains it. Also, the gene variant *APOE* ε4 was present exclusively in redescriptions describing subjects diagnosed with LMCI and AD.

We report several attributes, discovered during our analyses, for which we detected variations in levels connected to AD or discovered interesting subgroups of patients with significantly different distribution of values for a given attribute compared to CN subjects. Difference in distribution is measured with three different statistical tests: a) Anderson-Darling (ADT) test [51, 52], Kolmogorov-Smirnov (KST) test [53, 54] and Mann-Whitney U (MWUT) test [55]. For Anderson-Darling we perform two-sided test and report simulated

(p_s) and asymptotic (p_a) p-values, while for Kolmogorov-Smirnov and Mann-Whitney U test we report p-values, obtained by performing one-sided tests, and the observed direction of the shift of distribution. Alternative hypothesis (a), for one-tailed tests have two possible forms: a equals (=) less (l), or (a) = greater (g). Depending on the choice of statistical test, the alternative hypothesis have different meaning (explained in S17 File). Simulated p-value in ADT are obtained with default parameters (1000 simulations). Short motivation for the used statistical tests, providing references to implementations and meaning of the chosen alternative, for the used one-sided tests, is available in supplementary document S17 File. Tests of statistical significance of difference in distribution between one selected example group and a group of CN subjects for all mentioned attributes is displayed in Table 5. Information about attributes with statistically significant difference in distribution between AD/LMCI and CN subjects is reported in Table 6.

By observing redescriptions describing very homogeneous groups of subjects with high level of cognitive impairment (LMCI and AD), we discovered groups where testosterone levels (TSTSTRNT) were significantly decreased. Although some studies (e.g. Zhao et al. [56]) and meta-analyses showed no differences in plasma levels of testosterone between AD and matched controls (e.g. Xu et al. [57]), some studies, such as the one of Hogervorst et al. [58] and Lv et al. [59], found low free testosterone level to be an independent risk factor for AD. Plasma testosterone levels display circadian variation, peaking during sleep, and reaching a lowest level in the late afternoon, with a superimposed ultradian rhythm with pulses every 90 min reflecting the underlying rhythm of pulsatile luteinizing hormone (LH) secretion [60]. The increase in testosterone during sleep requires at least 3 hours of sleep with normal sleep architecture. However, since noradrenergic locus coeruleus and serotonergic dorsal raphe nucleus are among the first neurons affected by neurofibrillary tau pathology, their changes lead to the early and prominent deterioration of the sleep-wake cycle in AD (for a review, see šimić et al. [61]), which may add to a reduction of testosterone levels with advancing age. Experimental data obtained in animal models of AD suggest that low levels of testosterone increase Aβ and tau pathology through both androgen and estrogen pathways (testosterone is metabolized in the brain into androgen dihydrotestosterone, DHT, and 17β-estradiol, the E2 estrogen) [62, 63].

Table 5. Attributes analysed in this section with corresponding example redescription containing this attribute. For each selected attribute we present example redescription that describes subjects with statistically significant difference in attribute value distribution compared to a group of CN subjects.

Attribute	D	R	$\lvert E_{1,1}\rvert$	File	ADT		KST		MWUT	
					p_a	p_s	a	p	a	p
ANG2	D_3	R_{45}	46	S14	$4.1 \cdot 10^{-3}$	0	l	$2.7 \cdot 10^{-6}$	g	$4.7 \cdot 10^{-6}$
APOAII	D_3	R_{37}	55	S14	$7.3 \cdot 10^{-15}$	0	g	$1.7 \cdot 10^{-11}$	l	$4.2 \cdot 10^{-13}$
BNP	D_3	R_{96}	48	S14	$5.7 \cdot 10^{-3}$	0	l	0.02	g	0.15
CNTF	D_3	R_{56}	33	S13	0.03	0.03	l	0.02	g	0.02
TSTSTRNT	D_3	R_{85}	366	S15	$5 \cdot 10^{-6}$	0	g	0.002	l	0.05
INSULIN	D_3	R_{90}	5	S12	0.01	0.01	l	0.06	g	0.01
LEPTIN	D_3	R_{72}	24	S13	$9.4 \cdot 10^{-6}$	0	g	$5.1 \cdot 10^{-6}$	l	$7.4 \cdot 10^{-6}$
MCRPHMIF	D_3	R_{31}	6	S12	$9 \cdot 10^{-5}$	0	l	$4.2 \cdot 10^{-4}$	g	$2.3 \cdot 10^{-4}$
PAPP-A	D_3	R_{39}	327	S16	$3 \cdot 10^{-6}$	0.0	l	$4.8 \cdot 10^{-4}$	g	$1.8 \cdot 10^{-5}$
PPP	D_3	R_{43}	8	S12	$8.8 \cdot 10^{-3}$	0.13	l	0.02	g	0.008
SPARE_AD	D_3	R_{37}	155	S15	$1.2 \cdot 10^{-28}$	0.0	l	$2.2 \cdot 10^{-16}$	g	$2.2 \cdot 10^{-16}$

https://doi.org/10.1371/journal.pone.0187364.t005

Table 6. Analysed attributes with statistically significant difference in value distribution between groups of LMCI or AD patients and CN subjects.

Attribute	*D*	Type	ADT		KST		MWUT	
			p_a	p_s	a	*p*	a	*p*
APOAII	D_3	LMCI vs CN	$4.6 \cdot 10^{-11}$	0	g	$1.1 \cdot 10^{-9}$	l	$3.1 \cdot 10^{-10}$
	D_3	AD vs CN	$3.3 \cdot 10^{-7}$	0	g	$8.4 \cdot 10^{-5}$	l	$3.3 \cdot 10^{-7}$
APOB	D_3	AD vs CN	0.03	0.04	l	0.03	g	0.02
ANG2	D_3	LMCI vs CN	$2.6 \cdot 10^{-4}$	0	l	$4.8 \cdot 10^{-3}$	g	$1.5 \cdot 10^{-4}$
BNP	D_3	LMCI vs CN	$9.2 \cdot 10^{-8}$	0	l	$1.8 \cdot 10^{-5}$	g	$1.2 \cdot 10^{-6}$
	D_3	AD vs CN	$6 \cdot 10^{-7}$	0	l	$1.3 \cdot 10^{-5}$	g	$1.2 \cdot 10^{-6}$
FASL	D_3	LMCI vs CN	$3 \cdot 10^{-5}$	0	g	0.001	l	$2 \cdot 10^{-5}$
LEPTIN	D_3	LMCI vs CN	$1.2 \cdot 10^{-3}$	0	g	$6 \cdot 10^{-3}$	l	$4.7 \cdot 10^{-4}$
	D_3	AD vs CN	0.05	0.05	g	0.08	l	0.02
PAPP-A	D_3	LMCI vs CN	$7.2 \cdot 10^{-4}$	0.001	l	$1.3 \cdot 10^{-3}$	g	$3.4 \cdot 10^{-4}$
	D_3	AD vs CN	$6.1 \cdot 10^{-6}$	0	g	$1.1 \cdot 10^{-4}$	l	$8.3 \cdot 10^{-5}$
PPP	D_3	LMCI vs CN	$6.2 \cdot 10^{-3}$	0.005	l	0.009	g	0.003
	D_3	AD vs CN	$2.5 \cdot 10^{-3}$	0.001	l	0.007	g	$1.5 \cdot 10^{-3}$

https://doi.org/10.1371/journal.pone.0187364.t006

Unlike previous scarce data and negative correlation [64], we also found increased levels of ciliary neurotrophic factor (CNTF) in plasma in several redescriptions describing subjects with high level of cognitive impairment, together with decreased levels of leptin. The difference in distribution of leptin level between groups of AD/LMCI patients and CN subjects is significantly different (lower for AD and LMCI patients). This is in agreement with the results of Marwarha and Ghribi [65], showing that lower leptin levels detected in AD subjects can be a possible target for developing supplementation therapies for reducing the progression of AD. Some groups of subjects (such as R_{45} from S14 File) had significantly increased levels of plasma angiopoietin-2 (ANG2). This is in agreement with research by Thirumangalakudi et al. [66] and research by Grammas et al. [67], that revealed elevated expression of angiopietin-2 in the brains of AD subjects and the transgenic AD mice, respectively.

Increased levels of plasma brain natriuretic peptide (BNP) were found in several redescriptions containing subjects with severe cognitive impairment. Previous research [68] suggested that this peptide has more significant association with vascular dementia than with AD. This could suggest either that this group of subjects, described by redescriptions containing BNP attribute, suffered from both types of dementia (mixed dementia), or that these cases do not suffer from AD but indeed suffer from vascular dementia. Distributions of level of BNP are significantly different, in dataset D_3, between groups of LMCI/AD and CN subjects.

Finally, we also found alteration in plasma levels of several other attributes, whose relationship with AD has already been shown in the literature. These include increase in serum apolipoprotein B (APOB) [69], pancreatic polypeptide (PPP) [70, 71] and for very small groups, the increase of plasma insulin [72] and the CSF macrophage migration inhibitory factor (MCRPHMIF) [73] in AD brain. Fas (CD95) ligand (FASL) levels are found to be significantly decreased in LMCI patients compared to AD and CN subjects in our dataset. Levels are also lower in AD patients than in CN subjects but the difference is not statistically significant. Although one study suggests the upregulation of FASL in AD brain [74], the levels and variations seem to heavily depend on the part of the brain. For instance, FASL levels are found to be significantly decreased in hippocampus [75] in patients suffering from AD. Several groups of LMCI/AD patients with significantly lower levels of APOAII compared to the CN subjects were detected (which corresponds to research performed in [76, 77]). The difference in value

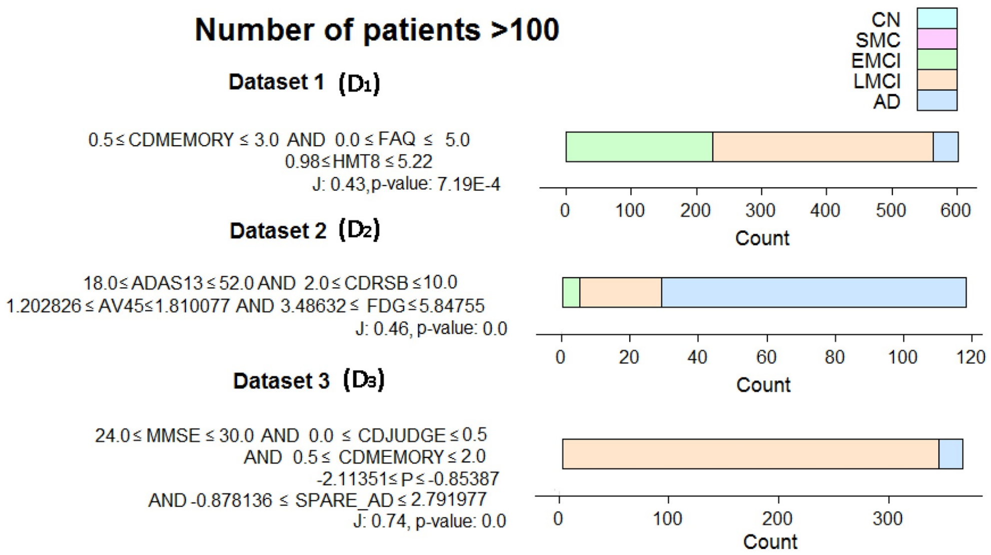


Fig 3. Example redescriptions (one for each dataset), each describing at least 100 subjects. All subjects described are diagnosed with EMCI, LMCI or AD. Attribute explanations can be seen in Tables 2 and 3 (P denotes PAPP-A and FDG denotes FDG-PET).

https://doi.org/10.1371/journal.pone.0187364.g003

distribution in dataset D_3 is significant between groups of LMCI/AD patients and CN subjects. Alterations in the levels of the PAPP-A attribute between CN subjects and LMCI/AD patients are very interesting (see Tables 5 and 6). The PAPP-A levels rise in LMCI subjects than drop significantly in AD subjects. This very property has been already detected in [78].

For each redescription set, we extracted one interesting, statistically significant redescription, and displayed its queries, along with the diagnosis distribution of the subjects described by this redescription (as shown in Fig 3).

The three redescriptions (as shown in Fig 3 from top to bottom) describe 602, 118 and 365 subjects, respectively with different proportion of EMCI, LMCI and AD diagnosis. They are statistically significant and describe 46%, 20% and 62% of all subjects with some level of cognitive impairment contained in the corresponding dataset. Their queries mostly contain well known attributes listed in Table 2 and in S1–S3 Files. The clinical attributes contained are memory score (CDMEMORY), Clinical Dementia Rating Scale Sum of Boxes (CDRSB), judgement and problem solving score (CDJUDGE), Alzheimer's Disease Assessment Scale (ADAS), Mini-Mental State Exam (MMSE). The biological attributes used contain neutrophils (HMT8), ^{18}F-florbetapir (AV45), ^{18}fluorodeoxyglucose—positron emission tomography (FDG-PET), Spatial Pattern of Abnormalities for Recognition of Early Alzheimer's disease (SPARE_AD) and Pregnancy-Associated Protein-A (PAPP-A) measurements.

Pairwise attribute association analysis based on co-occurrences

In this section, we present results of attribute association analyses based on attribute co-occurrences in queries of redescriptions contained in our redescription sets. To obtain these associations, we studied the attribute co-occurrence frequencies in redescriptions contained in redescription sets $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$. We focused on redescriptions describing subjects with some level of cognitive impairment. Co-occurrence frequencies were computed separately for pairs of attributes contained in views bio-bio, clin-clin and bio-clin, where *bio* denotes the view containing biological and *clin* denotes the view containing clinical attributes. Finally, we merged all redescriptions computed on all three datasets to obtain global information about pairwise attribute associations (set $\mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$). We do this for bio-bio, clin-clin and bio-clin combinations of views. Besides the associations, we also computed the pairwise attribute correlations, by using values of all subjects in the corresponding dataset for the selected pair of attributes, and the statistical significance of these correlations. For each attribute we performed the Kolmogorov-Smirnov test to assess if its values, for subjects contained in the dataset, follow normal distribution. If we obtained p-values smaller than 0.05 for both attributes in the pair, we computed Pearson correlation coefficient [79], otherwise we computed the Spearman's correlation coefficient [80] and the appropriate p-value of the corresponding significance test. Spearman's test was also used to compute correlations involving attributes with ordinal values.

A short list of top 5 pairwise associations (by co-occurrence) between attributes contained in the analysed datasets is provided in Tables 7, 8 and 9.

Table 7 shows high association between FDG-PET and the volume of the hippocampus, the entorhinal cortex, as well as an attribute related to the volume of the lateral ventricles. High association was also found between intracranial volume and creatine kinase levels (CKMB). This enzyme is present in greatest amounts in skeletal muscle, myocardium, and brain. The FDG-PET attribute often occurred in the same descriptive rules as the attribute measuring the level of vitamin B12 (BAT126). Administering of vitamin B12 is known to have beneficial effects on cognition when there is insufficient level of B12 in the organism [81, 82]. The incidence of AD increases with age and in fact, older adults often show deficiency of vitamin B12,

Table 7. The top five associations between pairs of biological attributes as measured by their co-occurrence in redescription queries. Attribute correlations for a redescription set $\mathcal{R}_i$ are computed on dataset D_i. *P* denotes the Pearson correlation coefficient and *S* denotes the Spearman's correlation coefficient. $\mathcal{R}_u = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$. Correlations for attribute pairs from the redescription set $\mathcal{R}_u$ are computed on the largest dataset containing both attributes.

Pairwise associations and correlations between biological attributes					
RS	**Attribute pair**	**Co-occurrence**	**Test**	**Correlation**	***p*-value**
$\mathcal{R}_1$	Hippocampus, FDG-PET	111	P	0.42	$<2.2 \cdot 10^{-16}$
	FDG-PET, Entorhinal	106	P	0.35	$<2.2 \cdot 10^{-16}$
	FDG-PET, Ventricles	52	S	−0.39	$<2.2 \cdot 10^{-16}$
	FDG-PET, ICV	46	S	−0.39	$<2.2 \cdot 10^{-16}$
	FDG-PET, AV45	42	S	−0.37	$<2.2 \cdot 10^{-16}$
$\mathcal{R}_2$	FDG.PET, Hippocampus	86	P	0.4	$<2.2 \cdot 10^{-16}$
	FDG-PET, Entorhinal	76	P	0.31	$<2.2 \cdot 10^{-16}$
	FDG-PET, AV45	52	S	−0.37	$<2.2 \cdot 10^{-16}$
	FDG-PET, RCT14	45	S	0.124	0.0003
	FDG-PET, BAT126	31	S	−0.007	0.42
$\mathcal{R}_3$	SPARE_AD, PAPP-A	66	S	−0.05	0.1
	SPARE_AD, Entorhinal	39	S	−0.51	$<2.2 \cdot 10^{-16}$
	PLMNRARC, PAPP-A	18	S	−0.05	0.14
	SPARE_AD, TNC	17	S	0.09	0.14
	PAPP-A, Entorhinal	15	S	0.08	0.039
$\mathcal{R}_u$	Hippocampus, FDG-PET	197	P	0.42	$<2.2 \cdot 10^{-16}$
	FDG-PET, Entorhinal	182	P	0.35	$<2.2 \cdot 10^{-16}$
	FDG-PET, AV45	94	S	−0.37	$<2.2 \cdot 10^{-16}$
	FDG-PET, Ventricles	68	S	−0.39	$<2.2 \cdot 10^{-16}$
	FDG-PET, ICV	67	S	−0.39	$<2.2 \cdot 10^{-16}$

https://doi.org/10.1371/journal.pone.0187364.t007

Table 8. The top five associations between pairs of clinical attributes as measured by their co-occurrence in redescription queries. Attribute correlations for a redescription set $\mathcal{R}_i$ are computed on dataset D_i. *P* denotes the Pearson correlation coefficient and *S* denotes the Spearman's correlation coefficient. $\mathcal{R}_u = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$. Correlations for attribute pairs from the redescription set $\mathcal{R}_u$ are computed on the largest dataset containing both attributes.

Pairwise associations and correlations between clinical attributes					
RS	Attribute pair	Co-occurrence	Test	Correlation	*p*-value
$\mathcal{R}_1$	ADAS13, RAVLT	52	S	−0.8	$<2.2 \cdot 10^{-16}$
	ADAS13, Q13SCORE	49	S	0.5	$<2.2 \cdot 10^{-16}$
	ADAS13, CDMEMORY	48	S	0.5	$<2.2 \cdot 10^{-16}$
	ADAS13, FAQ	45	S	0.67	$<2.2 \cdot 10^{-16}$
	RAVLT, CDMEMORY	43	S	−0.63	$<2.2 \cdot 10^{-16}$
$\mathcal{R}_2$	MOCA, ADAS13	60	S	−0.72	$<2.2 \cdot 10^{-16}$
	MOCA, EcogPtPlan	30	S	−0.28	$<2.2 \cdot 10^{-16}$
	MOCA, CDMEMORY	27	S	−0.58	$<2.2 \cdot 10^{-16}$
	ADAS13, MMSE	24	S	−0.64	$<2.2 \cdot 10^{-16}$
	ADAS13, CDRSB	23	S	0.66	$<2.2 \cdot 10^{-16}$
$\mathcal{R}_3$	ADAS13, CDMEMORY	60	S	0.76	$<2.2 \cdot 10^{-16}$
	MMSE, CDMEMORY	42	S	−0.73	$<2.2 \cdot 10^{-16}$
	ADAS13, CDRSB	34	S	0.76	$<2.2 \cdot 10^{-16}$
	ADAS13, MMSE	30	S	−0.71	$<2.2 \cdot 10^{-16}$
	FAQ, ADAS13	29	S	0.7	$<2.2 \cdot 10^{-16}$
$\mathcal{R}_u$	ADAS13, CDMEMORY	122	S	0.5	$<2.2 \cdot 10^{-16}$
	ADAS13, FAQ	82	S	0.67	$<2.2 \cdot 10^{-16}$
	ADAS13, CDRSB	79	S	0.72	$<2.2 \cdot 10^{-16}$
	ADAS13, RAVLT	77	S	−0.8	$<2.2 \cdot 10^{-16}$
	ADAS13, MMSE	77	S	−0.69	$<2.2 \cdot 10^{-16}$

https://doi.org/10.1371/journal.pone.0187364.t008

mainly due to the impaired vitamin B12 uptake in the gastrointestinal tract [83]. AD patients also have increased homocysteine levels in the blood. Since homocysteine is directly associated with brain atrophy, it is possible that vitamin B12 supplementation (that reduces homocysteine levels) can actually slow the progression of brain atrophy [81]. However, since meta-analyses failed to prove [84, 85] the connection of vitamin B12 supplementation with homocysteine levels and improved cognition, further studies should be conducted to resolve this issue. The correlation between FDG-PET and B12 values in our dataset was not statistically significant, though it may be more pronounced on a subset of subjects (for instance those above a certain age). It has been reported [86] that diagnosis based on FDG-PET can lead to false diagnosis of AD, where subjects can be cognitively normal or have cognitive impairment due to a reversible cause.

The clinical attributes ADAS, MOCA, MMSE, CDR, FAQ and RAVLT co-occurred frequently. Interestingly, the question number 13 (number of targets hit) from the ADAS test occurred very frequently in redescription queries. In this task, the participants are required to cross-out specific digits from a long list of digits. High frequency co-occurrences and corresponding correlations for all aforementioned attributes can be seen in Table 8.

There was also a strong association of the ADAS, CDR and MOCA clinical attributes with FDG-PET and SPARE_AD, the volume of the entorhinal cortex and the hippocampus, and other biological attributes (see Table 9). Correlations between these attributes were statistically significant. One of the most interesting associations is that between CDRSB and PAPP-A which is used in screening tests for Down syndrome. CDRSB and PAPP-A negatively correlated (−0.15) and the correlation was statistically significant at the significance level of 0.01.

Table 9. The top five associations between pairs of attributes consisting of a clinical and a biological attribute. The association is measured as their co-occurrence in redescription queries. Attribute correlations for a redescription set $\mathcal{R}_i$ are computed on dataset D_i. *P* denotes the Pearson correlation coefficient and *S* denotes the Spearman's correlation coefficient. $\mathcal{R}_u = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$. Correlations for attribute pairs from the redescription set $\mathcal{R}_u$ are computed on the largest dataset containing both attributes.

Pairwise associations and correlations between a biological and a clinical attribute					
RS	Attribute pair	Co-occurrence	Test	Correlation	*p*-value
$\mathcal{R}_1$	ADAS13, FDG	197	S	−0.58	$<2.2 \cdot 10^{-16}$
	ADAS13, Entorhinal	99	S	−0.49	$<2.2 \cdot 10^{-16}$
	ADAS13, Hippocampus	96	S	−0.54	$<2.2 \cdot 10^{-16}$
	ADAS11, FDG	79	S	−0.55	$<2.2 \cdot 10^{-16}$
	CDMEMORY, FDG	69	S	−0.49	$<2.2 \cdot 10^{-16}$
$\mathcal{R}_2$	ADAS13, FDG	142	S	−0.56	$<2.2 \cdot 10^{-16}$
	MOCA, FDG	124	S	0.49	$<2.2 \cdot 10^{-16}$
	ADAS13, Entorhinal	52	S	−0.38	$<2.2 \cdot 10^{-16}$
	MOCA, Hippocampus	44	S	0.45	$<2.2 \cdot 10^{-16}$
	RAVLT, FDG	42	S	0.48	$<2.2 \cdot 10^{-16}$
$\mathcal{R}_3$	ADAS13, SPARE_AD	131	S	0.68	$<2.2 \cdot 10^{-16}$
	CDMEMORY, SPARE_AD	110	S	0.72	$<2.2 \cdot 10^{-16}$
	CDRSB, SPARE_AD	58	S	0.7	$<2.2 \cdot 10^{-16}$
	MMSE, SPARE_AD	51	S	−0.62	$<2.2 \cdot 10^{-16}$
	CDRSB, PAPP-A	43	S	−0.15	0.0002
$\mathcal{R}_u$	ADAS13, FDG	339	S	−0.58	$<2.2 \cdot 10^{-16}$
	ADAS13, Entorhinal	171	S	−0.49	$<2.2 \cdot 10^{-16}$
	ADAS13, Hippocampus	136	S	−0.54	$<2.2 \cdot 10^{-16}$
	ADAS13, SPARE_AD	131	S	0.68	$<2.2 \cdot 10^{-16}$
	MOCA, FDG	124	S	0.49	$<2.2 \cdot 10^{-16}$

https://doi.org/10.1371/journal.pone.0187364.t009

Associations with PAPP-A. Motivated by the statistically significant association between PAPP-A and CDRSB, we used constraint-based redescription mining to create a new redescription set (on dataset D_3) by focusing only on redescriptions containing PAPP-A as one of the attributes in the redescription queries (corresponding redescription set is presented in supplementary document S16 File). The associations from this redescription set, containing 100 redescriptions, are presented in Table 10. Support sets of all constructed redescriptions contained both male and female subjects with diagnosis LMCI and AD.

The associations presented in Table 10 show that PAPP-A occurs frequently in redescription queries together with the clinical tests CDMEMORY, CDRSB, MMSE and ADAS13. Correlations between PAPP-A and all these attributes were statistically significant at the significance level of 0.01. Interestingly, SPARE_AD and PAPP-A occurred in every redescription from the redescription set obtained with constraint-based redescription mining. As noted earlier, the correlation between these two attributes was not statistically significant when measured for all subjects in the dataset. However, the correlation (Spearman's $\rho = -0.096$) was statistically significant (with $p = 0.026$) when measured for subjects with AD and LMCI at the significance level of 0.05. The fact that every redescription in the set obtained with constraint-based redescription mining described exclusively subjects with AD and LMCI possibly explains the high frequency of association between those attributes and necessitates further exploration of the role of PAPP-A in AD and LMCI. Additionally, we found an interesting association between PAPP-A and two other biological attributes: the volume of the entorhinal

Table 10. The top four associations of PAPP-A with other attributes based on attribute pair occurrences in redescription queries obtained by using constraint-based redescription mining on dataset D_3. *S* denotes Spearman's correlation coefficient. The produced redescription set contains 100 different redescriptions.

Associations of PAPP-A with biological attributes				
Attribute pair	**Co-occurrence**	**Test**	**Correlation**	***p*-value**
SPARE_AD, PAPP-A	100	S	−0.05	0.1
Fusiform, PAPP-A	21	S	0.11	0.01
Entorhinal, PAPP-A	20	S	0.08	0.039
Hippocampus, PAPP-A	13	S	0.01	0.4
Associations of PAPP-A with clinical attributes				
Attribute pair	**Co-occurrence**	**Test**	**Correlation**	***p*-value**
CDMEMORY, PAPP-A	85	S	−0.11	0.0034
CDRSB, PAPP-A	51	S	−0.15	0.00019
MMSE, PAPP-A	49	S	0.13	0.00088
ADAS13, PAPP-A	42	S	−0.11	0.0056

https://doi.org/10.1371/journal.pone.0187364.t010

cortex and the volume of the fusiform gyrus (Fusiform). Correlations between PAPP-A and these biological attributes were statistically significant at the significance level of 0.05.

Discussion

The redescription mining approach to segmenting high-dimensional datasets offers several advantages over classical clustering, subgroup discovery and association mining, such as the capability to generate relevant equivalence associations among combinations of attributes. We performed redescription mining experiments on three different datasets, created by extracting different sets of attributes from the ADNI database, and measured the redescription accuracy and the level of homogeneity (in terms of level of cognitive impairment) of the subjects described by each redescription. Basically, the main aim of our study has been to differentiate between cognitively normal subjects and those with some level of cognitive impairment, using clinical and biological attributes potentially related to AD. Our experiments over the constructed datasets were deliberately split into different support ranges in terms of subjects described with redescriptions to allow extracting general and specific, relevant AD-related information.

In this study, we found a number of surprisingly large and homogeneous groups and many smaller, more specific subgroups of subjects that are described with informative redescriptions, in a large extent confirming findings of previous works, corroborating some previously debatable findings or providing additional information about various attributes. After obtaining interesting associations with PAPP-A, we used the introduced extensions to the CLUS-RM algorithm to perform constraint-based redescription mining, allowing us to further explore associations of various attributes with PAPP-A. CLUS-RM is extended to perform fully automated constraint-based redescription mining on data containing either numerical, categorical attributes or missing values. In addition, it is equipped with soft and suggested CBRM capability, introduced in this work.

The clinical attribute CDR (CDMEMORY, CDGLOBAL and CDR-SB) was shown to be a very good attribute for differentiating CN subjects and subjects with some level of cognitive impairment. The gene variant *APOE ε*4 was associated with subjects with high level of cognitive impairment (LMCI and AD), whereas the biological attribute SPARE_AD was highly correlated with the subject's diagnosis.

Additionally, high association of ADAS, CDR, and MOCA clinical attributes with FDG-PET, SPARE_AD, and the volume of the entorhinal cortex and hippocampus were shown. When describing homogeneous groups of subjects with high level of cognitive impairment (LMCI and AD), the decrease of testosterone plasma levels, CNTF plasma levels and increase of BNP plasma levels were observed. Likewise, changes in other biological attributes previously reported as being altered in AD, such as increase in levels of serum apolipoprotein B, pancreatic polypeptide, plasma insulin and Fas (CD95) were found.

Finally, probably the most important finding of this study was the detection of altered levels of those biological attributes, for subjects with cognitive impairment, that could have potential as therapeutic targets in AD, namely decreased leptin and increased angiopoietin-2 plasma levels. Decreasing leptin levels have been suggested to alleviate AD-related cellular changes in rabbit organotypic slices [87] and in human neuroblastoma cell culture [88, 89], suggesting that lowered leptin levels detected in AD subjects can be a possible target for developing supplementation therapies for reducing the progression of AD. The finding of increased angiopoietin-2 plasma levels in AD patients is in accordance with the study of Thirumangalakudi et al. [66], who showed that angiopoietin-2 is expressed by AD, but not control-derived microvessels, supporting the idea of targeting the angiogenic changes in the microcirculation of the AD brain as a potential therapeutic approach in AD [67]. Altogether, analysing redescriptions from all three different datasets allowed finding many different associations. Some of these associations, such as SPARE_AD and PAPP-A are novel and require more in depth analysis with the supervision of domain experts. The correlation between SPARE_AD and PAPP-A was not statistically significant when computed for all subjects contained in the dataset $\mathcal{R}_3$, but it was statistically significant when computed only for subjects with AD and LMCI at the significance level of 0.05. PAPP-A showed significant correlation with the volume of the Fusiform gyrus and the volume of the Entorhinal cortex—both already known as being associated with AD [90, 91]. Further, PAPP-A had statistically significant correlation to the most widely used clinical cognitive tests: ADAS, Mini-Mental State Examination and Clinical Dementia Rating Sum of Boxes.

It has been shown [92] by measuring the reference intervals of PAPP-A (in 52 healthy males and 74 healthy, non-pregnant women) that the reference intervals are <22.9 ng/mL for men and <33.6 ng/mL for non-pregnant women. PAPP-A levels of smokers were lower than that of non-smokers and there is a positive correlation between serum PAPP-A levels and subjects' age. The measured median value of PAPP-A in males 6.85 with the range [undetectable, 24, 40] ng/mL were significantly higher than the median of female subjects 3.4 with the measured range [undetectable, 36, 7] ng/ml. For both males and females, non-smokers had higher levels of PAPP-A than smokers. For males, the difference was statistically significant and for females, it was not. PAPP-A levels in pre-menopause women were lower than in the post-menopause women, however the difference was not statistically significant. In male subjects, the study found a significant correlation between subjects' age and the level of PAPP-A, however in female subjects this correlation was not statistically significant.

Our search (PubMed search on 3 March 2016.) by using the keywords *pappalysin-1/Pregnancy-associated plasma protein-A (PAPP-A)* and *Alzheimer's disease* revealed only one publication [93] that associates PAPP-A with depressive symptoms.

Results by Llano et al. [78] show that PAPP-A is among the most significant descriptors in plasma proteomic data for distinguishing between CN, MCI and AD patients by different supervised machine learning algorithms. We discovered associations between PAPP-A and cognitive status (LMCI, AD). These results demonstrate the importance of further study of PAPP-A as potential marker for early detection of AD.

Distribution analysis of PAPP-A values based on our data and those of Llano et al. [78] show that PAPP-A levels are increased in MCI and LMCI patients but are significantly decreased in subjects diagnosed with AD. Decrease in PAPP-A levels from LMCI to AD patients on our data is more pronounced in female than in male patients. The possible link between PAPP-A and AD related genes (*ABCA1*, *ABCG1*) discovered in Hu et al. [94] is explained by Tang et al. [95]. This publication discusses the role of PAPP-A in pathogenesis of atherosclerosis through its inhibition of liver X receptors α (LXRα) through the insulin-like growth factor (IGF)-I-mediated signalling pathway, and negative regulation of expression of *ABCA1* and *ABCG1* genes—all significantly associated with AD [94]. Although LXR are best known as the key regulators of cholesterol metabolism and transport, LXR signaling has also been shown to have significant anti-inflammatory properties [96]. Various studies surveyed in štefulj et al. [96] implicate LXR in the pathogenesis, modulation, and therapy of AD.

Further potential association between PAPP-A and AD can be seen through study of patients suffering form type-2 diabetes. It has been shown [97] that patients suffering from type-2 diabetes also have significantly increased level of PAPP-A. Akter et al. [98] showed the potentially shared pathology of type-2 diabetes and AD, where some research (e.g. [99]), shows high influence of type-2 diabetes on the potential development of AD. Also, one study performed on mice [100] suggested that changes in the brain during AD can potentially cause diabetes.

Conclusion

The association of PAPP-A (previously known as pappalysin-1) with cognitive status is probably the most intriguing and novel finding of this study, as it has been scarcely investigated in this context.

PAPP-A was detected as a significant attribute in differentiating between CN, MCI and AD subjects [78] through use of different supervised machine learning algorithms. It has also been shown that it is significant in predicting the progression from MCI to AD, though none of the used subsets of attributes provided adequate predictions of progression between these two classes. High association of PAPP-A with depressive symptoms has already been demonstrated [93] by using the ensemble machine learning algorithm of Random Forests.

In our work, we detected important correlation between the attribute PAPP-A and the cognitive test CDRSB. By applying the newly developed constraint-based extensions of the CLUS-RM algorithm, we detected a larger number of attributes with statistically significant correlation with PAPP-A. In addition to CDRSB, we observed more clinical tests, such as MMSE and ADAS13, with statistically significant correlations with PAPP-A. Interesting and significant correlations were also observed with the biological attributes: volume of the Fusiform gyrus and volume of the Entorhinal cortex both known as being associated with AD [90, 91] with the volume of Entorhinal cortex being significantly reduced even in the mild case of AD [91].

The high importance of our finding lies in the fact that PAPP-A is a metalloproteinase, already known to cleave insulin-like growth factor (IGF) binding proteins (IGFBPs). Perhaps even more importantly, since it also shares similar substrates with the A Disintegrin and Metalloproteinase (ADAM) family of enzymes (the main group of enzymes that act as α-secretase to physiologically cleave the amyloid precursor protein (APP) in the so-called non-amyloidogenic pathway [101]), it could be directly involved in the metabolism of the amyloid precursor protein (APP) in the very early stages of AD. Based on the above, the role of PAPP-A in AD should be investigated in greater details.

Supporting information

S1 File. Dataset D_1 attribute structure.
(TXT)

S2 File. Dataset D_2 attribute structure.
(TXT)

S3 File. Dataset D_3 attribute structure.
(TXT)

S4 File. Redescriptions obtained on D_1 with support in [5, 10] interval.
(TXT)

S5 File. Redescriptions obtained on D_1 with support in [11, 39] interval.
(TXT)

S6 File. Redescriptions obtained on D_1 with support in [40, 99] interval.
(TXT)

S7 File. Redescriptions obtained on D_1 with support in [100, 820] interval.
(TXT)

S8 File. Redescriptions obtained on D_2 with support in [5, 10] interval.
(TXT)

S9 File. Redescriptions obtained on D_2 with support in [11, 39] interval.
(TXT)

S10 File. Redescriptions obtained on D_2 with support in [40, 99] interval.
(TXT)

S11 File. Redescriptions obtained on D_2 with support in [100, 470] interval.
(TXT)

S12 File. Redescriptions obtained on D_3 with support in [5, 10] interval.
(TXT)

S13 File. Redescriptions obtained on D_3 with support in [11, 39] interval.
(TXT)

S14 File. Redescriptions obtained on D_3 with support in [40, 99] interval.
(TXT)

S15 File. Redescriptions obtained on D_3 with support in [100, 420] interval.
(TXT)

S16 File. Redescriptions obtained on D_3 by using constraint-based redescription mining with support larger than 100 subjects.
(TXT)

S17 File. Motivation and explanation of statistical tests used in this work.
(PDF)

S18 File. Pseudocode of the CLUS-RM algorithm that can use conjunction, negation and disjunction logical operator in redescription query construction and explanation of introduced constraint-based redescription mining extensions.
(PDF)

Acknowledgments

Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (http://adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data, but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

Author Contributions

Conceptualization: Matej Mihelčić, Nada Lavrač, Sašo Džeroski, Tomislav Šmuc.

Formal analysis: Matej Mihelčić.

Funding acquisition: Goran Šimić, Nada Lavrač, Sašo Džeroski, Tomislav Šmuc.

Investigation: Matej Mihelčić, Goran Šimić, Tomislav Šmuc.

Methodology: Matej Mihelčić, Tomislav Šmuc.

Software: Matej Mihelčić.

Supervision: Nada Lavrač, Tomislav Šmuc.

Validation: Goran Šimić.

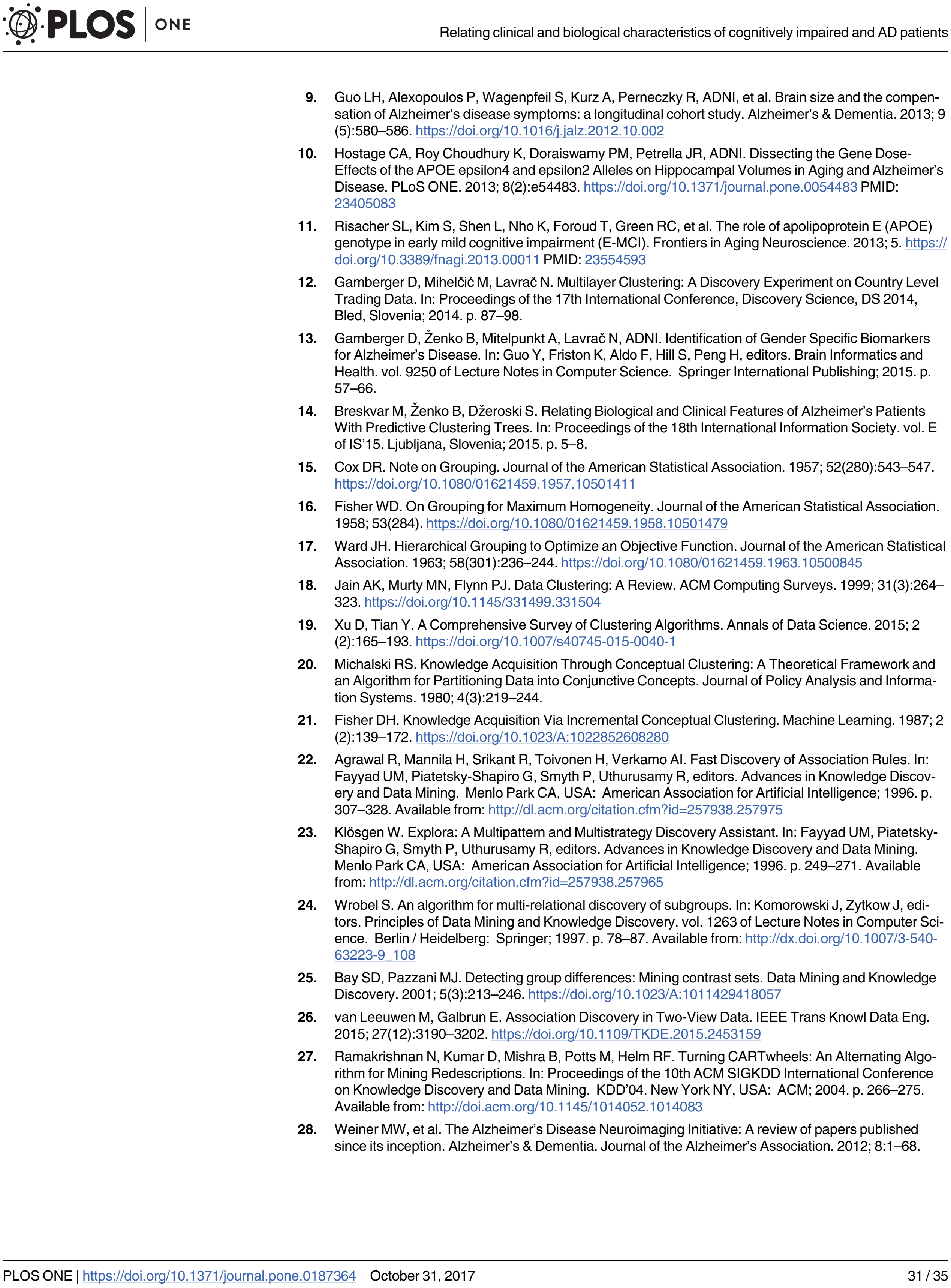
Visualization: Matej Mihelčić.

Writing – original draft: Matej Mihelčić, Goran Šimić, Mirjana Babić Leko, Nada Lavrač, Sašo Džeroski, Tomislav Šmuc.

Writing – review & editing: Matej Mihelčić, Goran Šimić, Mirjana Babić Leko, Nada Lavrač, Sašo Džeroski, Tomislav Šmuc.

References

1. ADNI database: last access 18.08.2017. Available from: http://adni.loni.usc.edu/
2. Smith G E, Bondi M W. Mild Cognitive Impairment and Dementia. Oxford University Press; 2013.
3. Gamberger D, Ženko B, Mitelpunkt A, Lavrač N. Multilayer Clustering: Biomarker Driven Segmentation of Alzheimer's Disease Patient Population. In: Ortuño F, Rojas I, editors. Bioinformatics and Biomedical Engineering. vol. 9043 of Lecture Notes in Computer Science. Springer International Publishing; 2015. p. 134–145.
4. Adaszewski S, Dukart J, Kherif F, Frackowiak R, Draganski B, ADNI, et al. How early can we predict Alzheimer's disease using computational anatomy? Neurobiology of aging. 2013; 34(12):2815–2826. https://doi.org/10.1016/j.neurobiolaging.2013.06.015 PMID: 23890839
5. Liu X, Tosun D, Weiner MW, Schuff N, Initiative ADN, et al. Locally linear embedding (LLE) for MRI based Alzheimer's disease classification. NeuroImage. 2013;83:148–157. https://doi.org/10.1016/j.neuroimage.2013.06.033 PMID: 23792982
6. Dukart J, Mueller K, Barthel H, Villringer A, Sabri O, Schroeter ML, et al. Meta-analysis based SVM classification enables accurate detection of Alzheimer's disease across different clinical centers using FDG-PET and MRI. Psychiatry Research: Neuroimaging. 2013; 212(3):230–236. https://doi.org/10.1016/j.pscychresns.2012.04.007 PMID: 23149027
7. Doraiswamy P, Sperling R, Johnson K, Reiman E, Wong T, Sabbagh M, et al. Florbetapir F 18 amyloid PET and 36-month cognitive decline: a prospective multicenter study. Molecular Psychiatry. 2014; 19 (9):1044–1051. https://doi.org/10.1038/mp.2014.9 PMID: 24614494
8. Donovan NJ, Wadsworth LP, Lorius N, Locascio JJ, Rentz DM, Johnson KA, et al. Regional cortical thinning predicts worsening apathy and hallucinations across the Alzheimer disease spectrum. The American Journal of Geriatric Psychiatry. 2014; 22(11):1168–1179. https://doi.org/10.1016/j.jagp.2013.03.006 PMID: 23890751



9. Guo LH, Alexopoulos P, Wagenpfeil S, Kurz A, Perneczky R, ADNI, et al. Brain size and the compensation of Alzheimer's disease symptoms: a longitudinal cohort study. Alzheimer's & Dementia. 2013; 9 (5):580–586. https://doi.org/10.1016/j.jalz.2012.10.002

10. Hostage CA, Roy Choudhury K, Doraiswamy PM, Petrella JR, ADNI. Dissecting the Gene Dose-Effects of the APOE epsilon4 and epsilon2 Alleles on Hippocampal Volumes in Aging and Alzheimer's Disease. PLoS ONE. 2013; 8(2):e54483. https://doi.org/10.1371/journal.pone.0054483 PMID: 23405083

11. Risacher SL, Kim S, Shen L, Nho K, Foroud T, Green RC, et al. The role of apolipoprotein E (APOE) genotype in early mild cognitive impairment (E-MCI). Frontiers in Aging Neuroscience. 2013; 5. https://doi.org/10.3389/fnagi.2013.00011 PMID: 23554593

12. Gamberger D, Mihelčić M, Lavrač N. Multilayer Clustering: A Discovery Experiment on Country Level Trading Data. In: Proceedings of the 17th International Conference, Discovery Science, DS 2014, Bled, Slovenia; 2014. p. 87–98.

13. Gamberger D, Ženko B, Mitelpunkt A, Lavrač N, ADNI. Identification of Gender Specific Biomarkers for Alzheimer's Disease. In: Guo Y, Friston K, Aldo F, Hill S, Peng H, editors. Brain Informatics and Health. vol. 9250 of Lecture Notes in Computer Science. Springer International Publishing; 2015. p. 57–66.

14. Breskvar M, Ženko B, Džeroski S. Relating Biological and Clinical Features of Alzheimer's Patients With Predictive Clustering Trees. In: Proceedings of the 18th International Information Society. vol. E of IS'15. Ljubljana, Slovenia; 2015. p. 5–8.

15. Cox DR. Note on Grouping. Journal of the American Statistical Association. 1957; 52(280):543–547. https://doi.org/10.1080/01621459.1957.10501411

16. Fisher WD. On Grouping for Maximum Homogeneity. Journal of the American Statistical Association. 1958; 53(284). https://doi.org/10.1080/01621459.1958.10501479

17. Ward JH. Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association. 1963; 58(301):236–244. https://doi.org/10.1080/01621459.1963.10500845

18. Jain AK, Murty MN, Flynn PJ. Data Clustering: A Review. ACM Computing Surveys. 1999; 31(3):264–323. https://doi.org/10.1145/331499.331504

19. Xu D, Tian Y. A Comprehensive Survey of Clustering Algorithms. Annals of Data Science. 2015; 2 (2):165–193. https://doi.org/10.1007/s40745-015-0040-1

20. Michalski RS. Knowledge Acquisition Through Conceptual Clustering: A Theoretical Framework and an Algorithm for Partitioning Data into Conjunctive Concepts. Journal of Policy Analysis and Information Systems. 1980; 4(3):219–244.

21. Fisher DH. Knowledge Acquisition Via Incremental Conceptual Clustering. Machine Learning. 1987; 2 (2):139–172. https://doi.org/10.1023/A:1022852608280

22. Agrawal R, Mannila H, Srikant R, Toivonen H, Verkamo AI. Fast Discovery of Association Rules. In: Fayyad UM, Piatetsky-Shapiro G, Smyth P, Uthurusamy R, editors. Advances in Knowledge Discovery and Data Mining. Menlo Park CA, USA: American Association for Artificial Intelligence; 1996. p. 307–328. Available from: http://dl.acm.org/citation.cfm?id=257938.257975

23. Klösgen W. Explora: A Multipattern and Multistrategy Discovery Assistant. In: Fayyad UM, Piatetsky-Shapiro G, Smyth P, Uthurusamy R, editors. Advances in Knowledge Discovery and Data Mining. Menlo Park CA, USA: American Association for Artificial Intelligence; 1996. p. 249–271. Available from: http://dl.acm.org/citation.cfm?id=257938.257965

24. Wrobel S. An algorithm for multi-relational discovery of subgroups. In: Komorowski J, Zytkow J, editors. Principles of Data Mining and Knowledge Discovery. vol. 1263 of Lecture Notes in Computer Science. Berlin / Heidelberg: Springer; 1997. p. 78–87. Available from: http://dx.doi.org/10.1007/3-540-63223-9_108

25. Bay SD, Pazzani MJ. Detecting group differences: Mining contrast sets. Data Mining and Knowledge Discovery. 2001; 5(3):213–246. https://doi.org/10.1023/A:1011429418057

26. van Leeuwen M, Galbrun E. Association Discovery in Two-View Data. IEEE Trans Knowl Data Eng. 2015; 27(12):3190–3202. https://doi.org/10.1109/TKDE.2015.2453159

27. Ramakrishnan N, Kumar D, Mishra B, Potts M, Helm RF. Turning CARTwheels: An Alternating Algorithm for Mining Redescriptions. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD'04. New York NY, USA: ACM; 2004. p. 266–275. Available from: http://doi.acm.org/10.1145/1014052.1014083

28. Weiner MW, et al. The Alzheimer's Disease Neuroimaging Initiative: A review of papers published since its inception. Alzheimer's & Dementia. Journal of the Alzheimer's Association. 2012; 8:1–68.

29. Mihelčić M, Džeroski S, Lavrač N, Šmuc T. Redescription mining with multi-label Predictive Clustering Trees. In: Proceedings of the 4th workshop on New Frontiers in Mining Complex Patterns. NFMCP'15. Porto, Portugal; 2015. p. 86–97. Available from: http://www.di.uniba.it/~ceci/micFiles/NFMCP2015.pdf

30. Mihelčić M, Džeroski S, Lavrač N, Šmuc T. Redescription Mining with Multi-target Predictive Clustering Trees. In: Ceci M, Loglisci C, Manco G, Masciari E, Ras ZW, editors. New Frontiers in Mining Complex Patterns—4th International Workshop, NFMCP 2015, Held in Conjunction with ECML-PKDD 2015, Porto, Portugal, Revised Selected Papers. vol. 9607 of Lecture Notes in Computer Science. Springer; 2015. p. 125–143. Available from: http://dx.doi.org/10.1007/978-3-319-39315-5_9

31. Zaki MJ, Ramakrishnan N. Reasoning About Sets Using Redescription Mining. In: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. KDD'05. New York, NY, USA: ACM; 2005. p. 364–373. Available from: http://doi.acm.org/10.1145/1081870.1081912

32. R Core Team. R: A Language and Environment for Statistical Computing; 2014. Available from: http://www.R-project.org/

33. ADNI1 procedures manual: last access 22.08.2017.;. Available from: http://www.adni-info.org/Scientists/doc/ADNI_GeneralProceduresManual.pdf

34. ADNIGO procedures manual: last access 22.08.2017.;. Available from: http://www.adni-info.org/Scientists/doc/ADNI_GO_Procedures_Manual_06102011.pdf

35. ADNI2 procedures manual: last access 22.08.2017.;. Available from: https://adni.loni.usc.edu/wp-content/uploads/2008/07/adni2-procedures-manual.pdf

36. Galbrun E. Methods for Redescription mining [Ph.D. dissertation]. University of Helsinki; 2013.

37. Galbrun E, Miettinen P. From black and white to full color: extending redescription mining outside the Boolean world. Statistical Analysis and Data Mining. 2012; 5(4):284–303. https://doi.org/10.1002/sam.11145

38. Zinchenko T. Redescription Mining Over non-Binary Data Sets Using Decision Trees, M.Sc. thesis [MSc dissertation]. Universität des Saarlandes Saarbrücken. Germany; 2014.

39. Parida L, Ramakrishnan N. Redescription Mining: Structure Theory and Algorithms. In: Veloso MM, Kambhampati S, editors. AAAI. AAAI Press / The MIT Press; 2005. p. 837–844. Available from: http://dblp.uni-trier.de/db/conf/aaai/aaai2005.html#ParidaR05

40. Gallo A, Miettinen P, Mannila H. Finding Subgroups having Several Descriptions: Algorithms for Redescription Mining. In: SDM. SIAM; 2008. p. 334–345. Available from: http://dblp.uni-trier.de/db/conf/sdm/sdm2008.html#GalloMM08

41. Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and Regression Trees. Monterey, CA: Wadsworth and Brooks; 1984.

42. Blockeel H, De Raedt L. Top-down Induction of First-order Logical Decision Trees. Artificial Intelligence. 1998; 101(1-2):285–297. https://doi.org/10.1016/S0004-3702(98)00034-4

43. Kocev D, Vens C, Struyf J, Dzeroski S. Tree ensembles for predicting structured outputs. Pattern Recognition. 2013; 46(3):817–833. https://doi.org/10.1016/j.patcog.2012.09.023

44. Mihelčić M, Džeroski S, Lavrač N, Šmuc T. Redescription mining augmented with random forest of multi-target predictive clustering trees. Journal of Intelligent Information Systems. 2017; p. 1–34.

45. Mihelčić M, Džeroski S, Lavrač N, Šmuc T. A framework for redescription set construction. Expert Systems with Applications. 2017; 68:196–215. https://doi.org/10.1016/j.eswa.2016.10.012

46. Piccart B. Algorithms for Multi-target Learning [Ph.D. dissertation]. Katholieke Universiteit Leuven. Belgium; 2012.

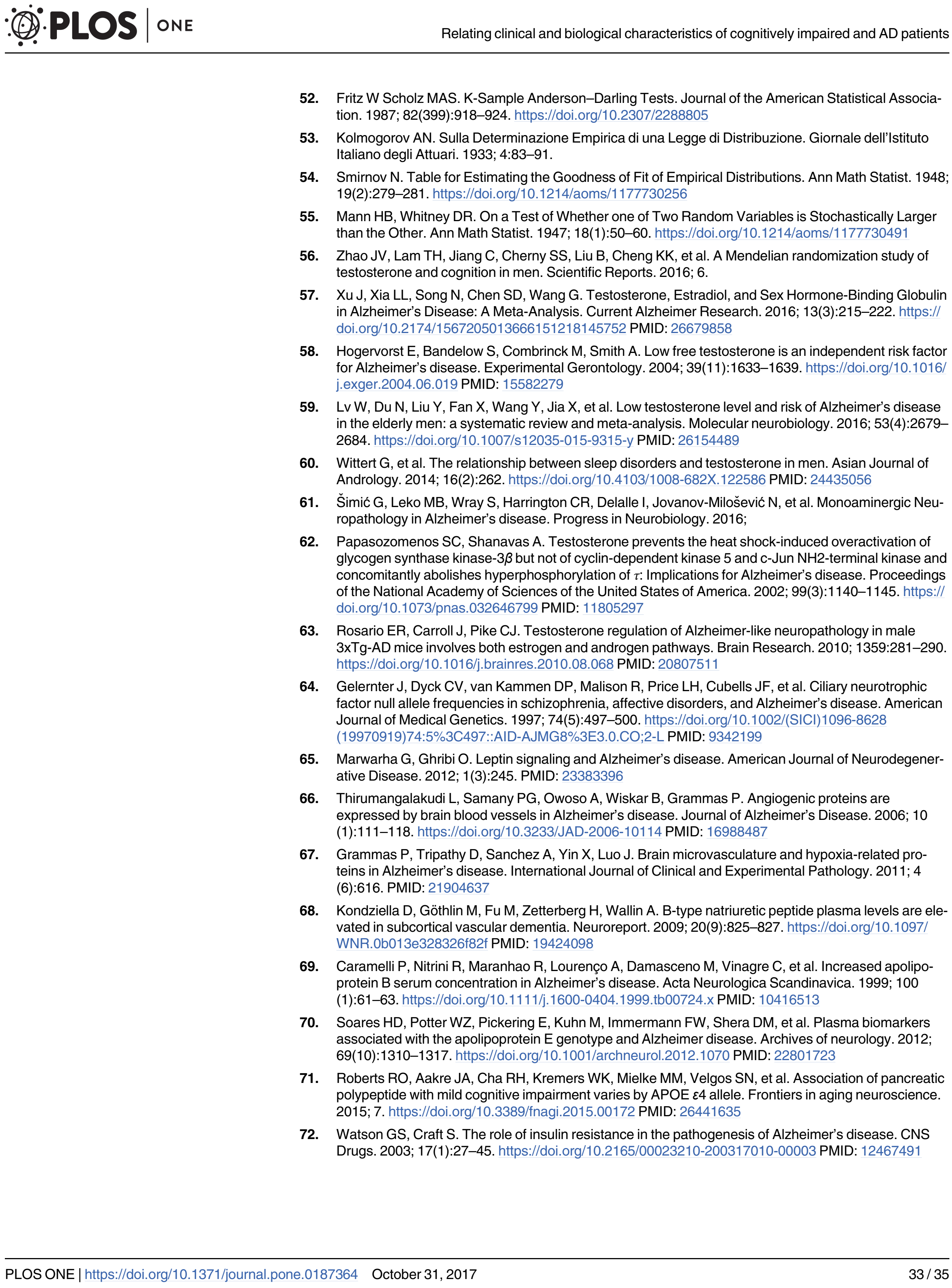
47. Galbrun E, Miettinen P. Siren: An Interactive Tool for Mining and Visualizing Geospatial Redescriptions. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD'12. New York, NY, USA: ACM; 2012. p. 1544–1547. Available from: http://doi.acm.org/10.1145/2339530.2339776

48. Mihelcic M, Smuc T. InterSet: Interactive Redescription Set Exploration. In: Discovery Science—19th International Conference, DS 2016, Bari, Italy, October 19-21, 2016, Proceedings; 2016. p. 35–50. Available from: http://dx.doi.org/10.1007/978-3-319-46307-0_3

49. Aho T, Ženko B, Džeroski S, Elomaa T. Multi-target Regression with Rule Ensembles. J Mach Learn Res. 2012; 13(1):2367–2407.

50. Shannon CE. A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review. 2001; 5(1):3–55. https://doi.org/10.1145/584091.584093

51. Anderson AD W Theodore Darling. Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes. Ann Math Statist. 1952; 23(2):193–212. https://doi.org/10.1214/aoms/1177729437



52. Fritz W Scholz MAS. K-Sample Anderson–Darling Tests. Journal of the American Statistical Association. 1987; 82(399):918–924. https://doi.org/10.2307/2288805

53. Kolmogorov AN. Sulla Determinazione Empirica di una Legge di Distribuzione. Giornale dell'Istituto Italiano degli Attuari. 1933; 4:83–91.

54. Smirnov N. Table for Estimating the Goodness of Fit of Empirical Distributions. Ann Math Statist. 1948; 19(2):279–281. https://doi.org/10.1214/aoms/1177730256

55. Mann HB, Whitney DR. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. Ann Math Statist. 1947; 18(1):50–60. https://doi.org/10.1214/aoms/1177730491

56. Zhao JV, Lam TH, Jiang C, Cherny SS, Liu B, Cheng KK, et al. A Mendelian randomization study of testosterone and cognition in men. Scientific Reports. 2016; 6.

57. Xu J, Xia LL, Song N, Chen SD, Wang G. Testosterone, Estradiol, and Sex Hormone-Binding Globulin in Alzheimer's Disease: A Meta-Analysis. Current Alzheimer Research. 2016; 13(3):215–222. https://doi.org/10.2174/1567205013666151218145752 PMID: 26679858

58. Hogervorst E, Bandelow S, Combrinck M, Smith A. Low free testosterone is an independent risk factor for Alzheimer's disease. Experimental Gerontology. 2004; 39(11):1633–1639. https://doi.org/10.1016/j.exger.2004.06.019 PMID: 15582279

59. Lv W, Du N, Liu Y, Fan X, Wang Y, Jia X, et al. Low testosterone level and risk of Alzheimer's disease in the elderly men: a systematic review and meta-analysis. Molecular neurobiology. 2016; 53(4):2679–2684. https://doi.org/10.1007/s12035-015-9315-y PMID: 26154489

60. Wittert G, et al. The relationship between sleep disorders and testosterone in men. Asian Journal of Andrology. 2014; 16(2):262. https://doi.org/10.4103/1008-682X.122586 PMID: 24435056

61. Šimić G, Leko MB, Wray S, Harrington CR, Delalle I, Jovanov-Milošević N, et al. Monoaminergic Neuropathology in Alzheimer's disease. Progress in Neurobiology. 2016;

62. Papasozomenos SC, Shanavas A. Testosterone prevents the heat shock-induced overactivation of glycogen synthase kinase-3*β* but not of cyclin-dependent kinase 5 and c-Jun NH2-terminal kinase and concomitantly abolishes hyperphosphorylation of *τ*: Implications for Alzheimer's disease. Proceedings of the National Academy of Sciences of the United States of America. 2002; 99(3):1140–1145. https://doi.org/10.1073/pnas.032646799 PMID: 11805297

63. Rosario ER, Carroll J, Pike CJ. Testosterone regulation of Alzheimer-like neuropathology in male 3xTg-AD mice involves both estrogen and androgen pathways. Brain Research. 2010; 1359:281–290. https://doi.org/10.1016/j.brainres.2010.08.068 PMID: 20807511

64. Gelernter J, Dyck CV, van Kammen DP, Malison R, Price LH, Cubells JF, et al. Ciliary neurotrophic factor null allele frequencies in schizophrenia, affective disorders, and Alzheimer's disease. American Journal of Medical Genetics. 1997; 74(5):497–500. https://doi.org/10.1002/(SICI)1096-8628(19970919)74:5%3C497::AID-AJMG8%3E3.0.CO;2-L PMID: 9342199

65. Marwarha G, Ghribi O. Leptin signaling and Alzheimer's disease. American Journal of Neurodegenerative Disease. 2012; 1(3):245. PMID: 23383396

66. Thirumangalakudi L, Samany PG, Owoso A, Wiskar B, Grammas P. Angiogenic proteins are expressed by brain blood vessels in Alzheimer's disease. Journal of Alzheimer's Disease. 2006; 10(1):111–118. https://doi.org/10.3233/JAD-2006-10114 PMID: 16988487

67. Grammas P, Tripathy D, Sanchez A, Yin X, Luo J. Brain microvasculature and hypoxia-related proteins in Alzheimer's disease. International Journal of Clinical and Experimental Pathology. 2011; 4(6):616. PMID: 21904637

68. Kondziella D, Göthlin M, Fu M, Zetterberg H, Wallin A. B-type natriuretic peptide plasma levels are elevated in subcortical vascular dementia. Neuroreport. 2009; 20(9):825–827. https://doi.org/10.1097/WNR.0b013e328326f82f PMID: 19424098

69. Caramelli P, Nitrini R, Maranhao R, Lourenço A, Damasceno M, Vinagre C, et al. Increased apolipoprotein B serum concentration in Alzheimer's disease. Acta Neurologica Scandinavica. 1999; 100(1):61–63. https://doi.org/10.1111/j.1600-0404.1999.tb00724.x PMID: 10416513

70. Soares HD, Potter WZ, Pickering E, Kuhn M, Immermann FW, Shera DM, et al. Plasma biomarkers associated with the apolipoprotein E genotype and Alzheimer disease. Archives of neurology. 2012; 69(10):1310–1317. https://doi.org/10.1001/archneurol.2012.1070 PMID: 22801723

71. Roberts RO, Aakre JA, Cha RH, Kremers WK, Mielke MM, Velgos SN, et al. Association of pancreatic polypeptide with mild cognitive impairment varies by APOE *ε*4 allele. Frontiers in aging neuroscience. 2015; 7. https://doi.org/10.3389/fnagi.2015.00172 PMID: 26441635

72. Watson GS, Craft S. The role of insulin resistance in the pathogenesis of Alzheimer's disease. CNS Drugs. 2003; 17(1):27–45. https://doi.org/10.2165/00023210-200317010-00003 PMID: 12467491

73. Bacher M, Deuster O, Aljabari B, Egensperger R, Neff F, Jessen F, et al. The role of macrophage migration inhibitory factor in Alzheimer's disease. Molecular Medicine. 2010; 16(3-4):116. https://doi.org/10.2119/molmed.2009.00123 PMID: 20200619

74. Su JH, Anderson AJ, Cribbs DH, Tu C, Tong L, Kesslack P, et al. Fas and Fas Ligand are associated with neuritic degeneration in the AD brain and participate in β-amyloid-induced neuronal death. Neurobiology of Disease. 2003; 12(3):182–193. https://doi.org/10.1016/S0969-9961(02)00019-0 PMID: 12742739

75. Ferrer I, Puig B, Krupinski J, Carmona M, Blanco R. Fas and Fas ligand expression in Alzheimer's disease. Acta neuropathologica. 2001; 102(2):121–131. PMID: 11563626

76. Song F, Poljak A, Crawford J, Kochan NA, Wen W, Cameron B, et al. Plasma Apolipoprotein Levels Are Associated with Cognitive Status and Decline in a Community Cohort of Older Individuals. PLOS ONE. 2012; 7(6):1–11. https://doi.org/10.1371/journal.pone.0034078

77. Ma C, Li J, Bao Z, Ruan Q, Yu Z. Serum levels of ApoA1 and ApoA2 are associated with cognitive status in older men. BioMed research international. 2015; 2015. https://doi.org/10.1155/2015/481621

78. Llano DA, Devanarayan V, Simon AJ, ADNI. Evaluation of plasma proteomic data for Alzheimer disease state classification and for the prediction of progression from mild cognitive impairment to Alzheimer disease. Alzheimer Disease & Associated Disorders. 2013; 27(3):233–243. https://doi.org/10.1097/WAD.0b013e31826d597a

79. Pearson K. Note on regression and inheritance in the case of two parents. Proceedings of the Royal Society of London. 1895; 58(347-352):240–242. https://doi.org/10.1098/rspl.1895.0041

80. Spearman C. The Proof and Measurement of Association Between Two Things. American Journal of Psychology. 1904; 15:88–103. https://doi.org/10.2307/1412159

81. Smith AD, Smith SM, de Jager CA, Whitbread P, Johnston C, Agacinski G, et al. Homocysteine-Lowering by B Vitamins Slows the Rate of Accelerated Brain Atrophy in Mild Cognitive Impairment: A Randomized Controlled Trial. PLoS ONE. 2010; 5(9):1–10. https://doi.org/10.1371/journal.pone.0012244

82. Douaud G, Refsum H, de Jager CA, Jacoby R, Nichols TE, Smith SM, et al. Preventing Alzheimer's disease-related gray matter atrophy by B-vitamin treatment. Proceedings of the National Academy of Sciences of the United States of America. 2013; 110(23):9523–9528. https://doi.org/10.1073/pnas.1301816110 PMID: 23690582

83. Gröber U, Kisters K, Schmidt J. Neuroenhancement with vitamin B12–underestimated neurological significance. Nutrients. 2013; 5(12):5031–5045. https://doi.org/10.3390/nu5125031 PMID: 24352086

84. Ford AH, Almeida OP. Effect of homocysteine lowering treatment on cognitive function: a systematic review and meta-analysis of randomized controlled trials. Journal of Alzheimer's Disease. 2012; 29 (1):133–149. PMID: 22232016

85. Clarke R, Bennett D, Parish S, Lewington S, Skeaff M, Eussen SJ, et al. Effects of homocysteine lowering with B vitamins on cognitive aging: meta-analysis of 11 trials with cognitive data on 22,000 individuals. The American Journal of Clinical Nutrition. 2014; 100(2):657–666. https://doi.org/10.3945/ajcn.113.076349 PMID: 24965307

86. Shipley SM, Frederick MC, Filley CM, Kluger BM. Potential for misdiagnosis in community-acquired PET scans for dementia. Neurology: Clinical Practice. 2013; 3(4):305–312.

87. Marwarha G, Dasari B, Prasanthi JR, Schommer J, Ghribi O. Leptin reduces the accumulation of Aβ and phosphorylated tau induced by 27-hydroxycholesterol in rabbit organotypic slices. Journal of Alzheimer's Disease. 2010; 19(3):1007–1019. https://doi.org/10.3233/JAD-2010-1298 PMID: 20157255

88. Marwarha G, Dasari B, Ghribi O. Endoplasmic reticulum stress-induced CHOP activation mediates the down-regulation of leptin in human neuroblastoma SH-SY5Y cells treated with the oxysterol 27-hydroxycholesterol. Cellular Signalling. 2012; 24(2):484–492. https://doi.org/10.1016/j.cellsig.2011.09.029 PMID: 21983012

89. Marwarha G, Raza S, Meiers C, Ghribi O. Leptin attenuates BACE1 expression and amyloid-β genesis via the activation of SIRT1 signaling pathway. Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease. 2014; 1842(9):1587–1595. https://doi.org/10.1016/j.bbadis.2014.05.015

90. Chang YT, Huang CW, Chen NC, Lin KJ, Huang SH, Chang WN, et al. Hippocampal amyloid burden with downstream fusiform gyrus atrophy correlate with face matching task scores in early stage Alzheimer's disease. Frontiers in aging neuroscience. 2016; 8. https://doi.org/10.3389/fnagi.2016.00145

91. Juottonen K, Laakso M, Insausti R, Lehtovirta M, Pitkänen A, Partanen K, et al. Volumes of the entorhinal and perirhinal cortices in Alzheimer's disease. Neurobiology of aging. 1998; 19(1):15–22. https://doi.org/10.1016/S0197-4580(98)00007-4 PMID: 9562498

92. Coskun A, Serteser M, Duran S, Inal TC, Erdogan BE, Ozpinar A, et al. Reference interval of pregnancy-associated plasma protein-A in healthy men and non-pregnant women. Journal of Cardiology. 2013; 61(2):128–131. https://doi.org/10.1016/j.jjcc.2012.09.007 PMID: 23159209

93. Arnold S, Xie S, Leung Y, Wang L, Kling M, Han X, et al. Plasma biomarkers of depressive symptoms in older adults. Translational Psychiatry. 2012; 2(1):e65. https://doi.org/10.1038/tp.2011.63 PMID: 22832727

94. Hu YS, Xin J, Hu Y, Zhang L, Wang J. Analyzing the genes related to Alzheimer's disease via a network and pathway-based approach. Alzheimer's Research & Therapy. 2017; 9(1):29. https://doi.org/10.1186/s13195-017-0252-z

95. Tang SL, Chen WJ, Yin K, Zhao GJ, Mo ZC, Lv YC, et al. PAPP-A negatively regulates ABCA1, ABCG1 and SR-B1 expression by inhibiting LXR*α* through the IGF-I-mediated signaling pathway. Atherosclerosis. 2012; 222(2):344–354. https://doi.org/10.1016/j.atherosclerosis.2012.03.005 PMID: 22503545

96. Štefulj J, Panzenboeck U, Hof PR, Šimić G. Pathogenesis, modulation, and therapy of Alzheimer's disease: A perspective on roles of liver-X receptors. Translational Neuroscience. 2013; 4(3):349–356.

97. Heidari B, Fotouhi A, Sharifi F, Mohammad K, Pajouhi M, Paydary K, et al. Elevated serum levels of pregnancy-associated plasma protein-A in type 2 diabetics compared to healthy controls: associations with subclinical atherosclerosis parameters. Acta Medica Iranica. 2015; 53(7):395–402. PMID: 26520625

98. Akter K, Lanza EA, Martin SA, Myronyuk N, Rua M, Raffa RB. Diabetes mellitus and Alzheimer's disease: shared pathology and treatment? British journal of clinical pharmacology. 2011; 71(3):365–376. https://doi.org/10.1111/j.1365-2125.2010.03830.x PMID: 21284695

99. Stanley M, Macauley SL, Holtzman DM. Changes in insulin and insulin signaling in Alzheimer's disease: cause or consequence? Journal of Experimental Medicine. 2016; p. jem–20160493.

100. Plucinska K, Dekeryte R, Koss D, Shearer K, Mody N, Riedel G, et al. Neuronal Human BACE1 Knock-in Induces Systemic Diabetes in Mice. In: Diabetes. vol. 65. Amer. Diabetes Assoc. 1701 N Beauregard st, Alexandria, VA 22311-1717 USA; 2016. p. A430–A430.

101. Kuhn PH, Colombo AV, Schusser B, Dreymueller D, Wetzel S, Schepers U, et al. Systematic substrate identification indicates a central role for the metalloprotease ADAM10 in axon targeting and synapse function. eLife. 2016; 5:e12748. https://doi.org/10.7554/eLife.12748 PMID: 26802628

Chapter 9

Software Availability

In this chapter we discuss the availability of software implementing algorithms and tools that constitute scientific contributions presented in this thesis.

9.1 Software Related to CLUS-RM and Redescription Set Optimization

The CLUS-RM algorithm (described in Sections 4.3 and 4.4) with redescription set optimization by redescription extraction[1] and by redescription exchange[2] (described in Sections 5.2 and 5.3) and the constraint-based redescription mining capability (described in Sections 4.5 and 5.4) is available in ClowdFlows [125] as a ClowdFlows widget. The complete code repository of widgets containing our library for redescription mining is available on GitHub[3]. The source code of the library for redescription mining and redescription set optimization is also available on GitHub[4].

We present a ClowdFlows workflow enabling the use of CLUS-RM algorithm, redescription set optimization procedure and the CLUS-RM capabilities for constraint-based redescription mining. The workflow is presented in Figure 9.1.

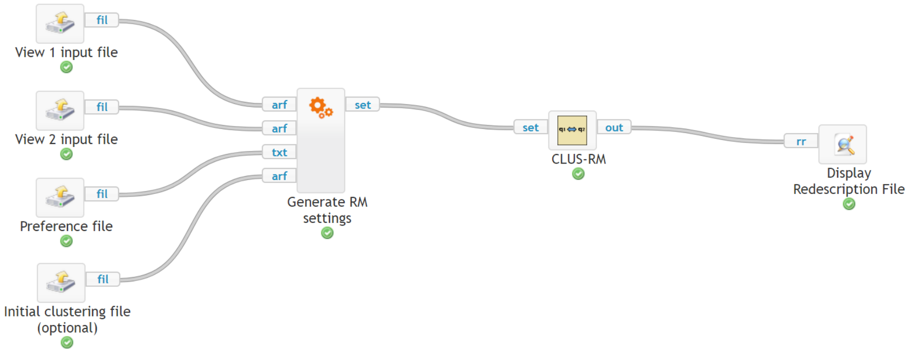


Figure 9.1: ClowdFlows workflow enabling redescription mining with the CLUS-RM algorithm and redescription set optimization.

[1] http://www.clowdflows.org/workflow/11552/

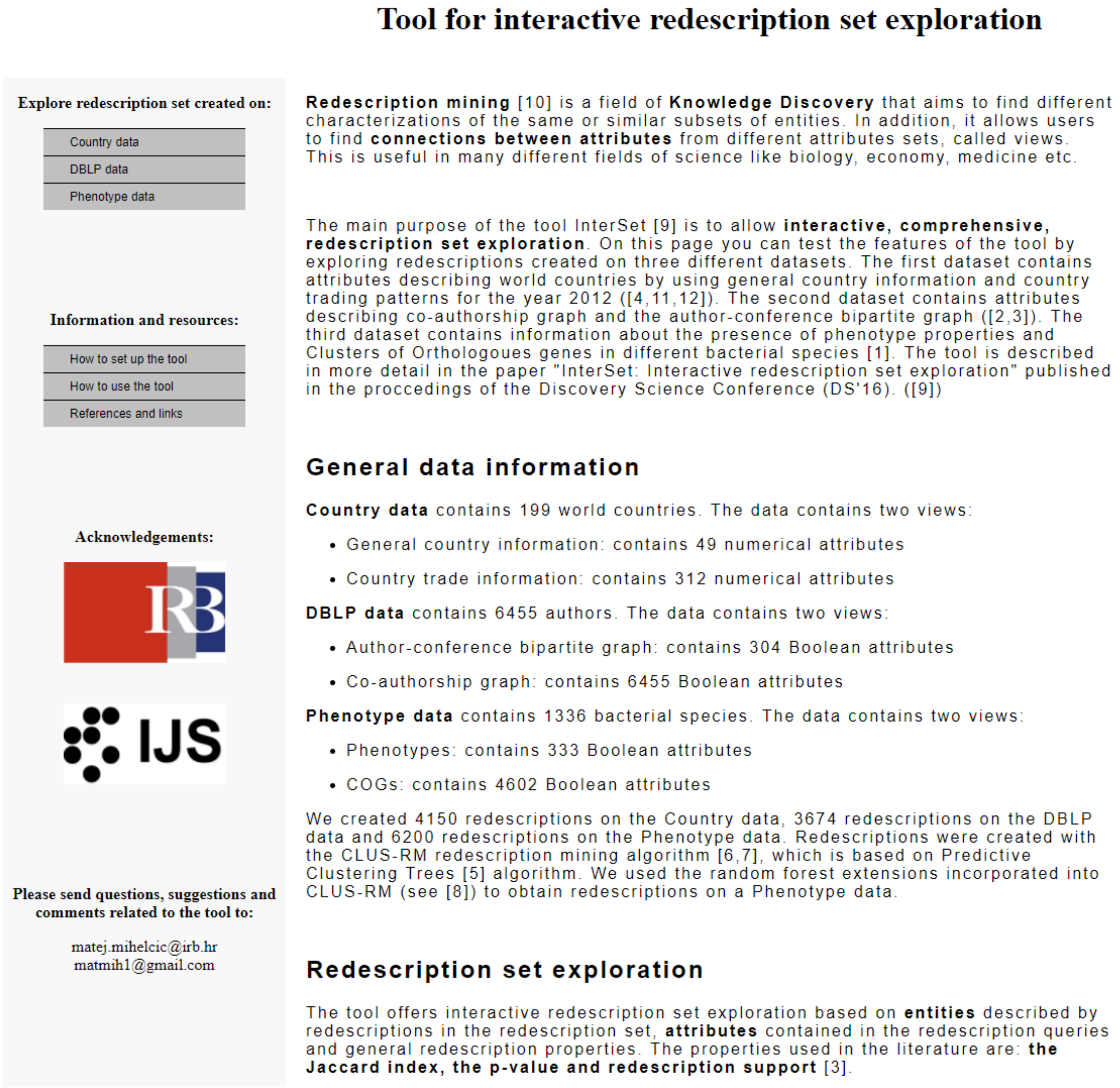
[2] http://www.clowdflows.org/workflow/11549/

[3] https://github.com/matmih/CLUS-RM-widgets-for-ClowdFlows

[4] https://github.com/matmih/CLUS-RM-library

9.2 Software Related to InterSet

InterSet is a web-based redescription set exploration tool (described in Section 6.3). The functionality of the tool can be tested on the Internet[5] (see Figure 9.2). The source code of the tool is freely available on GitHub[6].



Tool for interactive redescription set exploration

Explore redescription set created on:

Information and resources:

Acknowledgements:

Please send questions, suggestions and comments related to the tool to:

matej.mihelcic@irb.hr
matmih1@gmail.com

Redescription mining [10] is a field of **Knowledge Discovery** that aims to find different characterizations of the same or similar subsets of entities. In addition, it allows users to find **connections between attributes** from different attributes sets, called views. This is useful in many different fields of science like biology, economy, medicine etc.

The main purpose of the tool InterSet [9] is to allow **interactive, comprehensive, redescription set exploration**. On this page you can test the features of the tool by exploring redescriptions created on three different datasets. The first dataset contains attributes describing world countries by using general country information and country trading patterns for the year 2012 ([4,11,12]). The second dataset contains attributes describing co-authorship graph and the author-conference bipartite graph ([2,3]). The third dataset contains information about the presence of phenotype properties and Clusters of Orthologoues genes in different bacterial species [1]. The tool is described in more detail in the paper "InterSet: Interactive redescription set exploration" published in the proccedings of the Discovery Science Conference (DS'16). ([9])

General data information

Country data contains 199 world countries. The data contains two views:

- General country information: contains 49 numerical attributes
- Country trade information: contains 312 numerical attributes

DBLP data contains 6455 authors. The data contains two views:

- Author-conference bipartite graph: contains 304 Boolean attributes
- Co-authorship graph: contains 6455 Boolean attributes

Phenotype data contains 1336 bacterial species. The data contains two views:

- Phenotypes: contains 333 Boolean attributes
- COGs: contains 4602 Boolean attributes

We created 4150 redescriptions on the Country data, 3674 redescriptions on the DBLP data and 6200 redescriptions on the Phenotype data. Redescriptions were created with the CLUS-RM redescription mining algorithm [6,7], which is based on Predictive Clustering Trees [5] algorithm. We used the random forest extensions incorporated into CLUS-RM (see [8]) to obtain redescriptions on a Phenotype data.

Redescription set exploration

The tool offers interactive redescription set exploration based on **entities** described by redescriptions in the redescription set, **attributes** contained in the redescription queries and general redescription properties. The properties used in the literature are: **the Jaccard index, the p-value and redescription support** [3].

Figure 9.2: InterSet home page available at `http://zel.irb.hr/interset/`.

[5] `http://zel.irb.hr/interset/`

[6] `https://github.com/matmih/InterSet`

Chapter 10

Conclusions and Future Work

Contributions described in this thesis are primarily related to the data mining field called redescription mining. Predictive Clustering Trees constitute a backbone of the developed redescription mining algorithm called CLUS-RM. They have been utilized to construct rules used for redescription construction and to guide the search between different views. We have shown that using PCTs enables producing a large number of highly accurate redescriptions. Various extensions of CLUS-RM, such as using random forest of Predictive Clustering Trees, conjunctive refinement and query minimization procedures, have been developed with the aim of increasing the overall quality of produced redescription sets. The experimental evaluation shows that the CLUS-RM approach is fully competitive with other state-of-the art solutions and that it even outperforms other approaches when only conjunction and literal level negation logical operators are used in a redescription query construction.

A redescription set construction procedure, based on multi-objective optimization, that allows creating a redescription set of user-defined size, has also been developed. Depending on the type of optimization process used, the redescription set can be iteratively improved with newly constructed redescriptions or constructed by extracting redescriptions from a larger set of redescriptions. The optimization procedure allows limiting the number of produced patterns and influencing the redescription set structure, through various redescription quality measures and user importance preferences. This technique also allows for using ensembles of redescription mining algorithms to create redescription sets of superior properties in a fully automated manner.

The CLUS-RM redescription mining algorithm has also been extended to a constraint-based redescription mining setting (thus enabling the user to define attribute level constraints containing an arbitrary amount of attributes) and with new modes of constraint-based redescription mining.

The process of redescription set exploration, described in this thesis and realized through the tool InterSet complements the existing approach Siren and offers techniques for exploration of different parts of the generated redescription set. It derives different statistical information from the redescription set in order to enhance the exploration process. This differs from previous approaches that were based on examination of individual redescriptions.

The CLUS-RM algorithm with constraint-based redescription mining extensions has been applied in the domain of medicine to redescribe patients suffering from different levels of cognitive impairment or Alzheimer's disease. This resulted in confirming many already known findings, corroborating some debatable findings and providing research hypothesis about one scarcely explored finding—that connecting Pregnancy-associated plasma protein A (PAPP-A), with a different level of cognitive impairment and Alzheimer's disease.

Redescription mining currently aims to find descriptions of various subsets of entities using multiple distinct views that are as accurate as possible (subsets of entities described by redescription queries have a large Jaccard index) and are hard to obtain by joining randomly obtained queries into a redescription. However, it is not entirely clear what is the relation between these measures and the true interestingness of produced redescriptions. Current definition of the task is very broad and includes potential discovery of predictive patterns but also anomalies, outliers, one-time events and many other types of knowledge. Each of these tasks may be very interesting in different applications, however evaluation measures and underlying algorithms will probably need to be tuned and developed further to yield satisfactory results. The end usage and definition of interestingness largely depend on the domain and research objectives.

Methods and techniques presented in this thesis have been developed to solve the task in its broadest, most general form. The main goal was to allow users to obtain knowledge of interest and allow easier exploration and analysis of this knowledge. Although we significantly extended the number of measures used for redescription evaluation, there is still a lot of work to be done in this direction. As has been shown in Appendix B, many additional tests can be performed to evaluate different aspects of redescriptions and redescription mining algorithms.

The CLUS-RM algorithm has many advantages but also some disadvantages and potential for further improvement. The main drawback of the CLUS-RM algorithm is computational complexity (quadratic on the number of entities and linear in the number of attributes, the same as other tree-based approaches). It currently produces many tests, though this number can be reduced with various optimisations. Application of disjunction operator can potentially be improved by using procedure similar to atomic updates instead of combining existing precomputed queries. Such improvements could change the distribution of support size in the output redescription set, which currently contains larger number of redescriptions of high accuracy but lower support (mostly containing attributes strongly connected with conjunction operators). The optimization by extraction redescription set procedure can be consuming with respect to the used memory and execution time if it is used in an inappropriate manner.

In future work, we intend to extend the CLUS-RM methodology to incorporate the information about network connectivity between entities contained in the data. This will enable finding redescriptions with some specific network property. Further extensions include developing techniques for mining redescriptions described by more than two views, using redescription mining to find interesting subgroups described by multiple views and incorporating the developed extensions into the InterSet tool. Tuning CLUS-RM algorithm to find redescriptions of different properties (predictive, generalizable, significant) is another interesting direction. It includes development of appropriate techniques for further evaluation of redescriptions. Comparisons of different redescription mining algorithms may include measures such as rand index, adjusted rand index, variation of information etc. This may also serve as a good measure to include redescription mining algorithms that produce very different redescription sets in an ensemble to produce sets of superior properties. Potential of using redescription mining for different tasks, such as dimensionality reduction, prediction, anomaly detection, feature construction should be explored.

Appendix A

Correspondence Between Redescription Set and Multi-Objective Optimization

We demonstrate the correspondence between our redescription set optimization and the standard multi-objective optimization setup. This is done through a bijective mapping between redescriptions and numerical vectors. For each redescription quality criteria, we construct a function that returns an equivalent value to the corresponding redescription criteria given a numerical vector corresponding to a selected redescription.

As can be seen in Chapter 3, redescription quality criteria operate on a domain of redescriptions, occasionally using information about a redescription set. Standard multi-objective formulation assumes functions whose domain is a subset of $\mathbb{R}^n,\ n \in \mathbb{N},\ k \geq 2$. Thus, we demonstrate that there is a mapping between the two optimization approaches.

We can transform the originally defined multi-objective optimization problem

$$\begin{aligned}&\min \{f_1(R), f_2(R),\ \dots\ , f_k(R)\}\\ &R \in \mathcal{R}\end{aligned}$$

where $f_i : \mathcal{R} \mapsto \mathbb{R}$, into to the standard multi-objective setting

$$\begin{aligned}&\min \{f'_1(\vec{x}), f'_2(\vec{x}),\ \dots\ , f'_k(\vec{x})\}\\ &\vec{x} \in S \subset \mathbb{R}^n\end{aligned}$$

where $f'_i : \mathbb{R}^n \mapsto \mathbb{R},\ S \subset \mathbb{R}^n$.

Given an input dataset containing $|E|$ entities and two views containing $|\mathcal{A}_1|$ and $|\mathcal{A}_2|$ attributes, each redescription $R = (q_1, q_2)$ is represented with a vector $\vec{v} \in \mathbb{R}^n$ as follows:

- $v_i = 1 \leftrightarrow e_i \in \mathit{supp}(q_1),\ 0$ otherwise, for $i \leq |E|$.
- $v_i = 1 \leftrightarrow e_{i-|E|} \in \mathit{supp}(q_2),\ 0$ otherwise, for $|E| < i \leq 2 \cdot |E|$.
- $v_i = 1 \leftrightarrow e_{i-2\cdot|E|} \in \mathit{supp}(R),\ 0$ otherwise, for $2 \cdot |E| < i \leq 3 \cdot |E|$.
- $v_i = |\mathit{attr}(q_1) \cap \{a_j\}| \leftrightarrow a_j \in \mathit{attrs}(q_1),\ 0$ otherwise, for $3 \cdot |E| < i \leq (3 \cdot |E| + |\mathcal{A}_1|),\ j \in 1, \dots,\ |\mathcal{A}_1|$.
- $v_i = |\mathit{attr}(q_2) \cap \{b_k\}| \leftrightarrow b_k \in \mathit{attrs}(q_2),\ 0$ otherwise, for $(3 \cdot |E| + |\mathcal{A}_1|) < i \leq (3 \cdot |E| + |\mathcal{A}_1| + |\mathcal{A}_2|),\ k \in 1, \dots,\ |\mathcal{A}_2|$.

We can define the quality functions of redescriptions represented as vectors, with previously defined transformation, with equivalent functionality to redescription quality functions defined in Chapter 3. Newly defined functions have a domain $D \subseteq \mathbb{R}^n,\ n \in \mathbb{N}$, where the set D is finite and the feasible objective region of the multi-objective optimization problem is a non-convex set.

For a given dataset containing $|E|$ entities and $|\mathcal{A}_1|,\ |\mathcal{A}_2|$ attributes, we define vectors $\vec{e^{(i,j)}}$ defined as $e_k^{(i,j)} = 1,\ i < k \leq j,\ 0$ otherwise.

We define a function $f_J : \mathbb{R}^n \mapsto [0,1]$ which computes the Jaccard index of a redescription, represented as a real-valued vector. It is defined as:

$$f_J(\vec{v}) = \frac{\vec{v} \cdot \vec{e^{(2\cdot|E|,3\cdot|E|)}}}{\vec{v} \cdot \vec{e^{(0,|E|)}} + \vec{v} \cdot \vec{e^{(|E|,2\cdot|E|)}} - \vec{v} \cdot \vec{e^{(2\cdot|E|,3\cdot|E|)}}}$$

We denote $n = \frac{\vec{e^{(0,|E|)}} \cdot \vec{e^{(0,|E|)}}}{2}$. Statistical significance $f_p(\vec{v}) : \mathbb{R}^n \mapsto [0,1]$ is defined as:

$$f_p(\vec{v}) = \sum_{i=\vec{v}\cdot\vec{e^{(2\cdot|E|,3\cdot|E|)}}}^{n} \binom{n}{i} \cdot (\frac{\vec{v} \cdot \vec{e^{(0,|E|)}} \cdot \vec{v} \cdot \vec{e^{(|E|,2\cdot|E|)}}}{n^2})^n \cdot (1 - \frac{\vec{v} \cdot \vec{e^{(0,|E|)}} \cdot \vec{v} \cdot \vec{e^{(|E|,2\cdot|E|)}}}{n^2})^{n-i}$$

We define a set $\mathcal{B} \in \mathcal{P}(\mathbb{R}^n)$ such that $\vec{v} \in \mathcal{B} \leftrightarrow R \in \mathcal{R}$ and $\vec{v}$ is a vector representation of R and use it in the definition of functions:
$f_{redScoreEl}(\vec{v}) : \mathbb{R}^n \times \mathcal{P}(\mathbb{R}^n) \mapsto [0,1]$, defined as:

$$f_{redScoreEl}(\vec{v},\mathcal{B}) = \frac{\sum_{i,\vec{e^{(2\cdot|E|,3\cdot|E|)}}_i \neq 0} v_i \cdot (f_{\mathcal{B}}(e^{(i-1,i)}) - f_{ind}(\vec{v}))}{\sum_{i,\vec{e^{(2\cdot|E|,3\cdot|E|)}}_i \neq 0} f_{\mathcal{B}}(e^{(i-1,i)})}$$

where $f_{\mathcal{B}} : \mathbb{R}^n \mapsto \mathbb{R}$ is defined as $f_{\mathcal{B}}(\vec{d}) = \sum_{\vec{v'} \in \mathcal{B}} \vec{v'} \cdot \vec{d}$. Function $f_{ind} : \mathbb{R}^n \times \mathcal{P}(\mathbb{R}^n) \mapsto \mathbb{R}$ is defined as: $f_{ind}(\vec{v},\mathcal{B}) = \delta_{\vec{v}}(\mathcal{B})$. Auxiliary function $f_{el}(j) : \mathbb{N} \mapsto \mathcal{B} \subseteq \mathbb{R}^n,\ f_{el}(j) = \vec{v_k},\ k = j$ allows defining function $f_{score_{elemSim}} : \mathbb{R}^n \times \mathcal{P}(\mathbb{R}^n) \mapsto [0,1]$, as:

$$f_{score_{elemSim}}(\vec{v},\mathcal{B}) = max_j(\frac{e^{(2\cdot|E|,3\cdot|E|)} \cdot (\vec{v} \odot f_{el}(j,\mathcal{B}))}{e^{(2\cdot|E|,3\cdot|E|)} \cdot \vec{v} + e^{(2\cdot|E|,3\cdot|E|)} \cdot f_{el}(j,\mathcal{B}) - e^{(2\cdot|E|,3\cdot|E|)} \cdot (\vec{v} \odot f_{el}(j))})$$

Functions computing redescription attribute redundancy can be computed analogously. Redescription set attribute/entity coverage measure can be defined using a function $f_{exist} : \mathbb{N} \times \mathcal{P}(\mathbb{R}^n) \mapsto \{0,1\}$ defined as:

$$f_{exist}(i,\mathcal{B}) = \begin{cases} 1 & ,\sum_{\vec{v}\in\mathcal{B}} v_i > 0 \\ 0 & ,\text{otherwise} \end{cases}$$

Function $f_{cov_u} : \mathbb{R}^n \times \mathcal{P}(\mathbb{R}^n) \mapsto \mathbb{R}$ is defined as:

$$f_{cov_u}(\vec{v},\mathcal{B}) = \frac{\sum_{i=1}^{|E|} v_i \cdot (1 - f_{exist}(i,\mathcal{B}))}{|E|}$$

We define a family of functions $f_{comp}^{(k)} : \mathbb{R}^n \mapsto [0,1],\ k \in \mathbb{N}$ defined as:

$$f_{comp}^{(k)}(\vec{v}) = \begin{cases} \frac{\vec{v}\cdot e^{(3\cdot|E|,3\cdot|E|+|\mathcal{A}|_1+|\mathcal{A}|_2)}}{k} & ,\vec{v} \cdot e^{(3\cdot|E|,3\cdot|E|+|\mathcal{A}|_1+|\mathcal{A}|_2)} < k \\ 1 & ,k \leq \vec{v} \cdot e^{(3\cdot|E|,3\cdot|E|+|\mathcal{A}|_1+|\mathcal{A}|_2)} \end{cases}$$

Appendix B

Additional evaluation of CLUS-RM

We evaluate the CLUS-RM algorithm and produced redescriptions with several machine learning (predictivity tests [38], generalizability tests) and statistical techniques (permutation tests [126], Bonferroni correction [127] and FDR correction [128]). We discuss potential benefits and drawbacks of using these techniques in redescription mining. Evaluations are performed using the Country dataset which contains 199 entities (see [19]).

Predictivity test of redescription mining algorithms was introduced by Zinchenko et al. [38] to test if three different redescription mining algorithms produce predictive redescriptions. Predictive redescriptions are those whose accuracy is not significantly reduced when evaluated on the dataset that is a superset of a training set.

One significant problem that arises while using this procedure is that it is not possible to perform stratified sampling. Performing uniform sampling and then testing redescription mining algorithms causes unnatural splits that reduce accuracy of these algorithms. Moreover, groups of entities that would have been described using the original data are discarded due to insufficient support size when data is split to train and test set. In their work, Zinchenko et al. [38] stratify the Bio dataset using prior knowledge of its internal structure. Such procedure is hard to do for every dataset (especially if true structure is not known).

Here, we introduce the tests of *generalizability* of redescription mining algorithms. These tests assess algorithm capabilities to create redescriptions that generalize good to unseen entities (not contained in the datasets used to create redescriptions). By its nature, this test is similar to evaluating performance of predictive models on a test set. The artefacts of uniform sampling are even more pronounced and reflected on the test accuracy of redescriptions in this setting compared to tests of predictivity. The number of artefacts produced by testing procedures can be reduced by appropriate experimental setup. For a dataset E containing $|E|$ entities, we denote the training set with E_{tr} and the test set with E_{ts}. These sets are obtained by performing uniform split of the original dataset so that $|E_{tr}| = x \cdot |E|$ and $|E_{ts}| = (1-x) \cdot |E|$. We are interested in finding a group size c for some fixed group of entities, such that, after data split, the probability of having at least one entity from this group in test set is very close to 1. This size can be estimated from $p_e = 1 - \frac{\binom{|E|-c}{(1-x)|E|}}{\binom{|E|}{(1-x)|E|}}$. Plots showing probabilities of obtaining at least one entity from a group g of size $|g| = c \geq 5$, in a test, given train/test splits of 60/40%, 70/30% and 80/20%, are demonstrated in Figure B.1. These plots show that the probability stabilizes and gets very close to 1 for $|g| \geq 30$.

Splits that cause a very small number of entities from a given group to be present in a test set are also problematic (because they may cause large variability of test accuracy

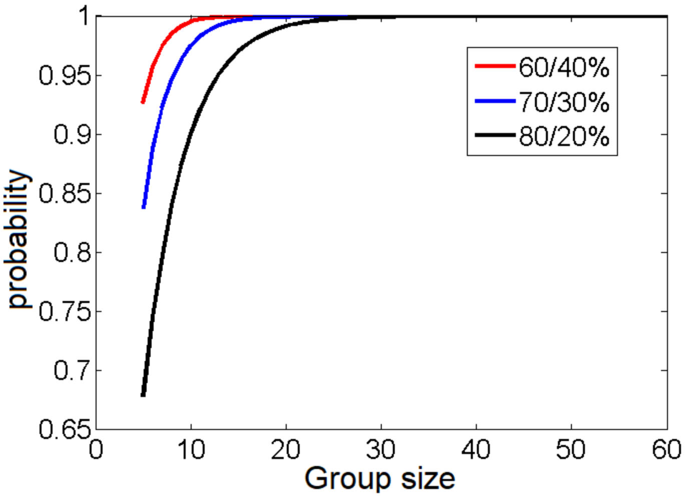


Figure B.1: Probabilities of selecting at least one entity from a group of entities of a size g in a test set with a pre-defined split for a Country dataset.

in redescriptions describing this group). For a targeted group of entities g of size $gs = |g|$, the probability of assigning at least s entities from g to the test set equals:

$$p_{eG}(s) = \sum_{i=s}^{min\{gs,|E_{ts}|\}} \binom{|E_{ts}|}{s} \cdot \frac{gs!}{(gs-i)!} \cdot \frac{(|E|-gs)!}{((|E|-gs)-|E_{ts}|+i)!} \cdot \frac{(|E|-|E_{ts}|)!}{|E|!}$$

We use this formula to compute the probability of having less than 10 entities in a test set for groups of size 30, 40 and 50 given a train/test split of 70/30%. The corresponding probabilities are 0.585, 0.162 and 0.021.

To evaluate capabilities of CLUS-RM to find predictive and generalizable redescriptions on a Country dataset, and to demonstrate aforementioned effects related to this type of evaluation, given uniform entity split, we create a set of 200 redescriptions with support intervals: $[10, 19]$, $[20, 39]$, $[40, 49]$ and $[50, 70]$. We plot a comparative boxplots showing distributions of Jaccard index obtained on a train set, the entire data set (assessing predictivity) and the test set (assessing generalizability). We count the number of redescriptions for which: a) there were no entities from test set contained in redescription support, b) the difference between the train Jaccard index and a test Jaccard index is larger than 0.2. CLUS-RM used minimal Jaccard index 0.5, maximal p-value 0.01 and 200 iterations with conjunction, negation and disjunction logical operators. Produced sets contain 200 redescriptions except the set corresponding to the last interval containing 130 discovered redescriptions. The results are presented in Figure B.2 and Table B.1.

Results presented in Figure B.2 show that the difference in redescription accuracy between train and test set slowly reduces as redescription support size increases.
Table B.1 demonstrates that large number of redescriptions with support size contained within first presented interval have no entities from test set contained within support set. As demonstrated, there is a non-zero probability that groups of size 10 to 20 will not have a representative in a test set. Given specific structure of Country dataset, where different subsets of Western European countries can be re-described with many different highly accurate redescriptions, such high number is not surprising. This potentially leads to hundreds of highly accurate redescriptions that describe countries exclusively contained within train set. Effects of errors due to small number of entities from a group contained in a test set can also be noticed from a column $|TD|$. As presented, with the increase of

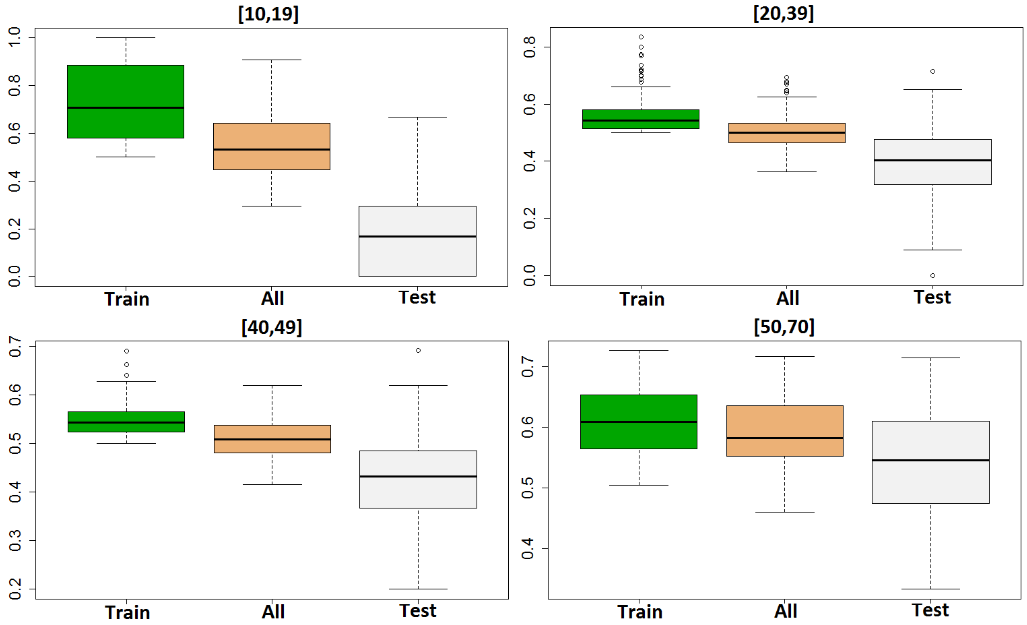


Figure B.2: Jaccard index comparison of redescription sets obtained using CLUS-RM algorithm, with different support parameters, on a train set, whole data set (testing predictivity) and a test set (testing generalizability) using 70/30% split.

Table B.1: The number of redescriptions $R \in \mathcal{R}$ with 0 entities from test set contained in their support set ($T0$) and the number of redescriptions for which $J(R)_{tr} - J(R)_{ts} > 0.2$ (TD).

$\mathcal{R}$	$\lvert\mathcal{R}\rvert$	$\lvert T0\rvert$	$\lvert TD\rvert$
$\mathcal{R}_{[10,19]}$	200	55	182
$\mathcal{R}_{[20,39]}$	200	2	62
$\mathcal{R}_{[40,49]}$	200	0	35
$\mathcal{R}_{[50,70]}$	135	0	12

minimal support size, the probability of such anomalies decreases significantly.

Pemutation tests were performed on a redescription set obtained from the Country dataset using 800 algorithm iterations, minimal Jaccard index 0.6 and other parameters as in previous experiments. The goal was to assess if CLUS-RM finds significant and accurate associations based on the properties of the underlying data (not solely due to distributions of attribute values). These tests were performed by randomly permuting values between entities in a Country dataset (so that associations between attributes are broken) and then using CLUS-RM algorithm on randomized dataset to find redescriptions with minimal support 5. The redescription produced on the original dataset was deemed significant if its accuracy was higher than predefined percentage of redescriptions produced on a randomized dataset. The main drawbacks of this approach are that it utilizes redescription mining algorithm being tested, it is time consuming and it is possible that no redescriptions, or insufficient number of redescriptions, can be found (in which case it is not possible to compute empirical p-values at a predefined level of significance). In such cases we consider all produced redescriptions as significant. Obtained results revealed that 192 out of 200 redescriptions have empirical p-value smaller or equal 0.01 and 196 smaller or equal to 0.05.

1 redescription has p-value larger or equal 0.1 meaning that at least 10% redescriptions obtained on the randomized dataset had larger accuracy than this redescription. This indicates that it may be possible to obtain redescriptions of this or smaller accuracy due to attribute value distributions occurring in the used dataset.

We also performed a permutation test in which we assessed the accuracy of redescriptions obtained on a Country dataset against 10000 randomized versions of this dataset. Randomized versions were obtained by randomly permuting attribute values among entities. For each redescription, we counted the number of times this redescription had accuracy larger on the original dataset then on the randomized versions (successes) and fails (cases when the opposite was true). The empirical p-value was assessed as a fraction of number of fails and the number of trials. The main drawback of this approach is that it requires intense computations. Obtained results showed that 121 of 200 redescriptions have $p_{emp} \leq 0.01$ and 155 have $p_{emp} \leq 0.05$. 34 redescriptions have $p_{emp} \geq 0.1$ showing that in at least 10% of the cases their accuracy emerged due to random arrangements of attribute values among entities.

Other form of randomization test, not performed but eligible for further work is to construct a random redescription mining algorithm and attempt to obtain redescriptions on a original dataset. This test could show if any of the discovered redescriptions by the redescription mining algorithm are easy to find at random from the original dataset.

We computed *Bonferroni* and *FDR* corrections for multiple hypothesis testing on the same redescription set on which we performed permutation tests. A major drawback of applying this method is that current approaches for computing p-values have a numerical limitation of $2 \cdot 10^{-17}$ which causes p-values to equal 0 on datasets containing more than 1000 entities. Another problem is determining what should be counted as a number of simultaneous tests, given the fact that different redescription mining algorithms work differently and that hypothesis are never assessed simultaneously. It is the iterative process in which many redescriptions are created but also discarded. CLUS-RM is even more specific with its redescription set optimization procedure. Due to this procedure and alternations, different redescriptions are added in a redescription set at different iterations. Corrected p-values may be a useful stopping criterion for CLUS-RM and other redescription mining algorithms. When the corrected p-values start to close to a predefined corrected value threshold, the alternation process could be terminated.

In this experiment, we compute the adjusted p-values for all redescriptions from a redescription set taking into account all hypothesis tests performed during CLUS-RM computations (which is highly conservative) and computing the adjusted p-values taking into account output redescription set size (which might be permissive since some redescriptions were compared with others during optimization process). We argue that the true estimate of corrected p-value is somewhere between these numbers.

The obtained results, when p-values were corrected with the total number of tests performed $(13, 255, 436)$, using Bonferroni correction, showed that 112 redescriptions have corrected p-value smaller or equal 0.01 and 152 smaller or equal 0.05. 4 redescriptions have corrected p-value larger or equal 0.1. Bonferroni correction is criticized for being to strict and having small power. When corrected using the size of output redescription set as a number of simultaneous tests, 198 redescriptions have corrected p-value less than 0.01 and 2 redescriptions have corrected p-value larger or equal 0.1. FDR corrected p-values differ significantly. When correcting using all hypothesis tests performed, the corrected p-values of 198 redescriptions is less or equal 0.01 whereas 2 redescriptions have corrected p-value larger or equal 0.1. When FDR is computed using only the size of output redescription set as number of simultaneous tests, the corresponding corrected p-value is smaller than 0.01 for all produced redescriptions.

References

[1] M. Goebel and L. Gruenwald, “A survey of data mining and knowledge discovery software tools,” *ACM SIGKDD Explorations Newsletter*, vol. 1, no. 1, pp. 20–33, 1999, ISSN: 1931-0145.

[2] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “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. From Data Mining to Knowledge Discovery: An Overview, pp. 1–34, ISBN: 0-262-56097-6.

[3] N. Ramakrishnan, D. Kumar, B. Mishra, M. Potts, and R. F. Helm, “Turning CARTwheels: An alternating algorithm for mining redescriptions,” in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA*, 2004, pp. 266–275.

[4] D. Gamberger, M. Mihelčić, and N. Lavrač, “Multilayer clustering: A discovery experiment on country level trading data,” in *Proceedings of Discovery Science - 17th International Conference, DS 2014, Bled, Slovenia, October 8-10, 2014.*, 2014, pp. 87–98.

[5] UNCTAD, *Unctad database*, Accessed October 2013, 2014. [Online]. Available: `http://unctad.org/en/Pages/Statistics.aspx`.

[6] WorldBank, *World bank database*, Accessed October 2013, 2014. [Online]. Available: `http://data.worldbank.org/`.

[7] R. J. Hijmans, S. E. Cameron, J. L. Parra, P. G. Jones, and A. Jarvis, “Very high resolution interpolated climate surfaces for global land areas,” *International Journal of Climatology*, vol. 25, no. 15, 2005. [Online]. Available: `www.worldclim.org`.

[8] A. Mitchell-Jones, *The Atlas of European Mammals*, ser. Poyser natural history. T & AD Poyser, 1999, ISBN: 9780856611308. [Online]. Available: `www.european-mammals.org`.

[9] E. Galbrun, “Methods for Redescription Mining,” PhD thesis, University of Helsinki, Finland, 2013.

[10] L. Parida and N. Ramakrishnan, “Redescription mining: Structure theory and algorithms,” in *Proceedings of the 20th National Conference on Artificial Intelligence, AAAI 2005*, Pittsburgh, Pennsylvania: AAAI Press, 2005, pp. 837–844, ISBN: 1-57735-236-x.

[11] M. J. Zaki and N. Ramakrishnan, “Reasoning about sets using redescription mining,” in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2005*, Chicago, Illinois, USA: ACM, 2005, pp. 364–373.

[12] A. Gallo, P. Miettinen, and H. Mannila, "Finding subgroups having several descriptions: Algorithms for redescription mining.," in *Proceedings of SIAM International Conference on Data Mining, SDM 2008*, SIAM, 2008, pp. 334–345.

[13] E. Galbrun and P. Miettinen, "From black and white to full color: Extending redescription mining outside the boolean world," *Statistical Analysis and Data Mining*, vol. 5, no. 4, pp. 284–303, 2012.

[14] T. Zinchenko, "Redescription mining over non-binary data sets using decision trees," Master's thesis, Universität des Saarlandes Saarbrücken, Germany, 2014.

[15] E. Galbrun and P. Miettinen, "Siren: An interactive tool for mining and visualizing geospatial redescriptions.," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2012*, Q. Yang, D. Agarwal, and J. Pei, Eds., ACM, 2012, pp. 1544–1547.

[16] H. Blockeel, L. D. Raedt, and J. Ramon, "Top-down induction of clustering trees," in *Proceedings of the Fifteenth International Conference on Machine Learning, ICML 1998*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 55–63, ISBN: 1-55860-556-8.

[17] D. Kocev, C. Vens, J. Struyf, and S. Džeroski, "Tree ensembles for predicting structured outputs," *Pattern Recognition*, vol. 46, no. 3, pp. 817–833, 2013.

[18] M. Mihelčić, S. Džeroski, N. Lavrač, and T. Šmuc, "Redescription mining augmented with random forest of multi-target predictive clustering trees," *Journal of Intelligent Information Systems*, pp. 1–34, 2017, In press.

[19] M. Mihelčić, S. Džeroski, N. Lavrač, and T. Šmuc, "Redescription mining with multi-target predictive clustering trees," in *Proceedings of the 4th International Workshop, New Frontiers in Mining Complex Patterns, NFMCP 2015, Held in conjunction with ECML-PKDD 2015, Porto, Portugal, September 7, 2015, Revised Selected Papers*, M. Ceci, C. Loglisci, G. Manco, E. Masciari, and Z. W. Ras, Eds. Cham: Springer International Publishing, 2016, pp. 125–143.

[20] M. Mihelčić, S. Džeroski, N. Lavrač, and T. Šmuc, "A framework for redescription set construction," *Expert Systems with Applications*, vol. 68, pp. 196–215, 2017, ISSN: 0957-4174.

[21] M. Mihelčić and T. Šmuc, "InterSet: Interactive redescription set exploration," in *Proceedings of Discovery Science: 19th International Conference, DS 2016, Bari, Italy, October 19–21, 2016*, T. Calders, M. Ceci, and D. Malerba, Eds. Cham: Springer International Publishing, 2016, pp. 35–50.

[22] M. Mihelčić, G. Šimić, M. Babić Leko, N. Lavrač, S. Džeroski, T. Šmuc, and for the Alzheimer's Disease Neuroimaging Initiative, "Using redescription mining to relate clinical and biological characteristics of cognitively impaired and alzheimer's disease patients," *PLOS ONE*, vol. 12, no. 10, pp. 1–35, 2017.

[23] ADNI, *ADNI database*, 2015. [Online]. Available: `http://adni.loni.usc.edu/`.

[24] DBLP, *DBLP dataset*, 2010. [Online]. Available: `http://dblp.uni-trier.de/db`.

[25] E. Galbrun and A. Kimmig, "Finding relational redescriptions," *Machine Learning*, vol. 96, no. 3, pp. 225–248, 2013.

[26] J. Fürnkranz, D. Gamberger, and N. Lavrač, *Foundations of Rule Learning.* Springer Publishing Company, Incorporated, 2014, ISBN: 3642430465, 9783642430466.

[27] C. Sammut and G. I. Webb, "Disjunctive normal form," in *Encyclopedia of Machine Learning.* Boston, MA: Springer US, 2010, pp. 289–289.

[28] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, "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. Fast Discovery of Association Rules, pp. 307–328, ISBN: 0-262-56097-6.

[29] J. Hipp, U. Güntzer, and G. Nakhaeizadeh, "Algorithms for association rule mining — a general survey and comparison," *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 1, pp. 58–64, 2000, ISSN: 1931-0145.

[30] M. Zhang and C. He, "Survey on association rules mining algorithms," in *Advancing Computing, Communication, Control and Management*, Q. Luo, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 111–118.

[31] M. J. Zaki and C.-J. Hsiao, "Charm: An efficient algorithm for closed itemset mining.," in *Proceedings of SIAM International Conference on Data Mining, SDM 2002*, R. L. Grossman, J. Han, V. Kumar, H. Mannila, and R. Motwani, Eds., SIAM, 2002, pp. 457–473.

[32] T. Calders and B. Goethals, "Mining all non-derivable frequent itemsets," in *Proceedings of Principles of Data Mining and Knowledge Discovery: 6th European Conference, PKDD 2002 Helsinki, Finland*, T. Elomaa, H. Mannila, and H. Toivonen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 74–86.

[33] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *ACM SIGMOD Record*, vol. 29, no. 2, pp. 1–12, 2000.

[34] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[35] H. Blockeel and L. D. Raedt, "Top-down induction of first-order logical decision trees.," *Artificial Intelligence*, vol. 101, no. 1-2, pp. 285–297, 1998.

[36] H. Blockeel, "Top-down induction of first order logical decision trees," PhD thesis, Katholieke Universiteit Leuven, Belgium, 1998.

[37] B. Ženko, "Learning Predictive Clustering Rules," PhD thesis, University of Ljubljana, Slovenia, 2007.

[38] T. Zinchenko, E. Galbrun, and P. Miettinen, "Mining predictive redescriptions with trees," in *Proceedings of IEEE International Conference on Data Mining Workshop, ICDMW 2015, Atlantic City, NJ, USA*, 2015, pp. 1672–1675.

[39] J. Kalofolias, E. Galbrun, and P. Miettinen, "From sets of good redescriptions to good sets of redescriptions," in *Proceedings of IEEE 16th International Conference on Data Mining, ICDM 2016, Barcelona, Spain*, 2016, pp. 211–220.

[40] N. Ramakrishnan and M. J. Zaki, "Redescription mining and applications in bioinformatics," *Biological Data Mining*, p. 561, 2009.

[41] R. G. Pearson and T. P. Dawson, "Predicting the impacts of climate change on the distribution of species: Are bioclimate envelope models useful?" *Global Ecology and Biogeography*, vol. 12, no. 5, pp. 361–371, 2003.

[42] E. Galbrun and P. Miettinen, "Analysing political opinions using redescription mining," in *Proceedings of 16th IEEE International Conference on Data Mining Workshops, ICDMW 2016*, 2016, pp. 422–427.

[43] M. van Leeuwen and E. Galbrun, "Association discovery in two-view data," in *Proceedings of 32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland*, 2016, pp. 1480–1481.

[44] D. R. Cox, "Note on grouping," *Journal of the American Statistical Association*, vol. 52, no. 280, pp. 543–547, 1957.

[45] W. D. Fisher, "On grouping for maximum homogeneity," *Journal of the American Statistical Association*, vol. 53, no. 284, 1958.

[46] J. H. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.

[47] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999, ISSN: 0360-0300.

[48] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, no. 2, pp. 165–193, 2015.

[49] R. S. Michalski, "Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts," *Journal of Policy Analysis and Information Systems*, vol. 4, no. 3, pp. 219–244, 1980.

[50] D. H. Fisher, "Knowledge acquisition via incremental conceptual clustering," *Machine Learning*, vol. 2, no. 2, pp. 139–172, 1987, ISSN: 0885-6125.

[51] W. Klösgen, "Explora: A multipattern and multistrategy discovery assistant," in *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds., Menlo Park, CA, USA: American Association for Artificial Intelligence, 1996, pp. 249–271.

[52] S. Wrobel, "An algorithm for multi-relational discovery of subgroups," in *Proceedings of Principles of Data Mining and Knowledge Discovery*, ser. Lecture Notes in Computer Science, J. Komorowski and J. Zytkow, Eds., vol. 1263, Berlin / Heidelberg: Springer, 1997, pp. 78–87.

[53] 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, ISSN: 1532-4435.

[54] F. Herrera, C. J. Carmona, P. González, and M. J. Jesus, "An overview on subgroup discovery: Foundations and applications," *Knowledge and Information Systems*, vol. 29, no. 3, pp. 495–525, 2010.

[55] G. Dong and J. Li, "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 1999*, San Diego, California, USA: ACM, 1999, pp. 43–52, ISBN: 1-58113-143-7.

[56] S. D. Bay and M. J. Pazzani, "Detecting group differences: Mining contrast sets," *Data Mining and Knowledge Discovery*, vol. 5, no. 3, pp. 213–246, 2001.

[57] D. Leman, A. Feelders, and A. Knobbe, "Exceptional model mining," in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, ECML-PKDD 2008*, Antwerp, Belgium: Springer-Verlag, 2008, pp. 1–16.

[58] S. Bickel and T. Scheffer, "Multi-view clustering," in *Proceedings of the 4th IEEE International Conference on Data Mining ICDM 2004, Brighton, UK*, 2004, pp. 19–26.

[59] H. Wang, F. Nie, and H. Huang, "Multi-view clustering and feature learning via structured sparsity," in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA*, 2013, pp. 352–360.

[60] S. Sun, "A survey of multi-view machine learning," *Neural Computing and Applications*, vol. 23, no. 7, pp. 2031–2038, Dec. 2013.

[61] M. Gönen and E. Alpayd, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2211–2268, Jul. 2011, ISSN: 1532-4435.

[62] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[63] N. Japkowicz and M. Shah, *Evaluating Learning Algorithms: A Classification Perspective.* New York, NY, USA: Cambridge University Press, 2011, ISBN: 0521196000, 9780521196000.

[64] N. Lavrač, P. Flach, and B. Zupan, "Rule evaluation measures: A unifying view," in *Proceedings of Inductive Logic Programming: 9th International Workshop, ILP 1999 Bled, Slovenia*, S. Džeroski and P. Flach, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 174–185.

[65] 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, ISSN: 1532-4435.

[66] T. Abudawood and P. Flach, "Evaluation measures for multi-class subgroup discovery," in *Proceedings of Machine Learning and Knowledge Discovery in Databases: European Conference, ECML-PKDD 2009, Bled, Slovenia*, W. Buntine, M. Grobelnik, D. Mladenić, and J. Shawe-Taylor, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 35–50.

[67] D. S. Moore and G. P. Mccabe, *Introduction to the Practice of Statistics*, Second edition. New York: Freeman, 1993.

[68] G. I. Webb, S. Butler, and D. Newlands, "On detecting differences between groups," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2003*, Washington, D.C.: ACM, 2003, pp. 256–265.

[69] P. K. Novak, N. Lavrač, D. Gamberger, and A. Krstačić, "CSM-SD: Methodology for contrast set mining through subgroup discovery," *Journal of Biomedical Informatics*, vol. 42, no. 1, pp. 113–122, 2009.

[70] X.-H. Huynh, "Interestingness Measures for Association Rules in a KDD Process: PostProcessing of Rules with ARQAT Tool," Theses, Université de Nantes, 2006.

[71] N. Bhargava and M. Shukla, "Survey of interestingness measures for association rules mining: Data mining, data science for business perspective," *International Journal of Computer Science and Information Technology & Security*, vol. 6, no. 2, 2016.

[72] J. L. Balcázar and F. Dogbey, "Evaluation of association rule quality measures through feature extraction," in *Proceedings of the International Symposium on Intelligent Data Analysis*, Springer Berlin Heidelberg, 2013, pp. 68–79.

[73] A. Ragel and a. Crémilleux, "Treatment of missing values for association rules," in *Proceedings of Research and Development in Knowledge Discovery and Data Mining, Second Pacific-Asia Conference, PAKDD 1998 Melbourne, Australia*, Springer, 1998, pp. 258–270.

[74] D. Gamberger and N. Lavrač, "Expert-guided subgroup discovery: Methodology and application," *Journal of Artificial Intelligence Research*, vol. 17, no. 1, pp. 501–527, 2002, ISSN: 1076-9757.

[75] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986, ISSN: 0885-6125.

[76] D. L. Zimmerman and N. Cressie, "Mean squared prediction error in the spatial linear model with estimated covariance parameters," *Annals of the Institute of Statistical Mathematics*, vol. 44, no. 1, pp. 27–43, 1992.

[77] B. Piccart, "Algorithms for multi-target learning," PhD thesis, Katholieke Universiteit Leuven, Belgium, 2012.

[78] D. Kroening and O. Strichman, *Decision Procedures: An Algorithmic Point of View*, 1st ed. Springer Publishing Company, Incorporated, 2008, ISBN: 3540741046, 9783540741046.

[79] R. M. Karp, "Reducibility among combinatorial problems," in *Proceedings of a symposium on the Complexity of Computer Computations, IBM Thomas J. Watson Research Center, Yorktown Heights, New York*, R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, Eds. Boston, MA: Springer US, 1972, pp. 85–103.

[80] A. Caprara, P. Toth, and M. Fischetti, "Algorithms for the set covering problem," *Annals of Operations Research*, vol. 98, no. 1, pp. 353–371, Dec. 2000.

[81] J. Branke, K. Deb, K. Miettinen, and R. Slowinski, Eds., *Multiobjective Optimization, Interactive and Evolutionary Approaches [outcome of Dagstuhl seminars].* Vol. 5252, Lecture Notes in Computer Science, Springer, 2008, ISBN: 978-3-540-88907-6.

[82] S. Greco, M. Ehrgott, and J. Figueira, *Multiple Criteria Decision Analysis: State of the Art Surveys*, ser. International Series in Operations Research & Management Science. Springer New York, 2016, ISBN: 9781493930944.

[83] R. Benayoun, J. De Montgolfier, J. Tergny, and O. Laritchev, "Linear programming with multiple objective functions: Step method (stem)," *Mathematical programming*, vol. 1, no. 1, pp. 366–375, 1971.

[84] P. Korhonen, S. Salo, and R. E. Steuer, "A Heuristic for Estimating Nadir Criterion Values in Multiple Objective Linear Programming," *Operations Research*, vol. 45, no. 5, pp. 751–757, 1997.

[85] H. Weistroffer, "Careful usage of pessimistic values is needed in multiple objectives optimization," *Operations Research Letters*, vol. 4, no. 1, pp. 23–25, 1985.

[86] K. Miettinen, *Nonlinear Multiobjective Optimization.* Kluwer Academic Publishers, Boston, 1999.

[87] K. Deb, S. Chaudhuri, and K. Miettinen, "Towards Estimating Nadir Objective Vector Using Evolutionary Approaches," in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO 2006*, Seattle, Washington, USA: ACM, 2006, pp. 643–650.

[88] K. Deb and K. Miettinen, "Nadir Point Estimation Using Evolutionary Approaches: Better Accuracy and Computational Speed Through Focused Search," in *Proceedings of the 19th International Conference on Multiple Criteria Decision Making, Auckland, New Zealand*, M. Ehrgott, B. Naujoks, T. J. Stewart, and J. Wallenius, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 339–354.

[89] K. Deb, K. Miettinen, and S. Chaudhuri, "Toward an Estimation of Nadir Objective Vector Using a Hybrid of Evolutionary and Local Search Approaches," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 6, pp. 821–841, 2010, ISSN: 1089-778X.

[90] Y. Sawaragi, H. Nakayama, and T. Tanino, *Theory of Multiobjective Optimization.* Elsevier, 1985, vol. 176.

[91] S. Gass and T. Saaty, "The computational algorithm for the parametric objective function," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 39–45, 1955.

[92] L. Zadeh, "Optimality and non-scalar-valued performance criteria," *IEEE Transactions on Automatic Control*, vol. 8, no. 1, pp. 59–60, Jan. 1963.

[93] Y. Haimes, L. Lasdon, and D. Wismer, "On a bicriterion formation of the problems of integrated system identification and system optimization," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-1, no. 3, pp. 296–297, 1971, ISSN: 0018-9472.

[94] V. Chankong and Y. Y. Haimes, *Multiobjective decision making: theory and methodology.* Courier Dover Publications, 2008.

[95] R. L. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* Cambridge University Press, 1993.

[96] P. C. Fishburn, "Exceptional paper—lexicographic orders, utilities and decision rules: A survey," *Management Science*, vol. 20, no. 11, pp. 1442–1471, 1974.

[97] A. Charnes, W. W. Cooper, and R. O. Ferguson, "Optimal estimation of executive compensation by linear programming," *Management Science*, vol. 1, no. 2, pp. 138–151, 1955.

[98] W. J. Baumol, "Management models and industrial applications of linear programming," *Naval Research Logistics Quarterly*, vol. 9, no. 1, pp. 63–64, 1962.

[99] M. Zeleny, "Compromise programming," in *Multiple Criteria Decision Making*, J. Cochrane and M. Zeleny, Eds., Columbia: University of South Carolina Press, 1973, pp. 262–301.

[100] A. Wierzbicki, M. Makowski, J. Wessels, *et al.*, *Model-Based Decision Support Methodology with Environmental Applications.* Kluwer Academic Dordrecht, The Netherlands, 2000.

[101] P. L. Yu, "A class of solutions for group decision problems," *Management Science*, vol. 19, no. 8, pp. 936–946, 1973.

[102] Y. Censor, "Pareto optimality in multiobjective problems," *Applied Mathematics and Optimization*, vol. 4, no. 1, pp. 41–59, Mar. 1977.

[103] I. Das and J. E. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems," *Structural and Multidisciplinary Optimization*, vol. 14, no. 1, pp. 63–69, 1997.

[104] B. Roy and V. Mousseau, "A theoretical framework for analysing the notion of relative importance of criteria," *Journal of Multi-Criteria Decision Analysis*, vol. 5, no. 2, pp. 145–159, 1996.

[105] R. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application*, ser. Wiley Series in Probability and Mathematical Statistics. Wiley, 1986.

[106] L. Tanner, "Selecting a text-processing system as a qualitative multiple criteria problem," *European Journal of Operational Research*, vol. 50, no. 2, pp. 179–187, 1991.

[107] H. Lakkaraju, S. H. Bach, and J. Leskovec, "Interpretable decision sets: A joint framework for description and prediction," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, San Francisco, California, USA: ACM, 2016, pp. 1675–1684, ISBN: 978-1-4503-4232-2.

[108] P. Miettinen, "Interactive data mining considered harmful (if done wrong)," in *ACM SIGKDD 2014 Full-day Workshop on Interactive Data Exploration and Analytics*, 2014, pp. 85–87.

[109] M. Boley, M. Mampaey, B. Kang, P. Tokmakov, and S. Wrobel, "One click mining: Interactive local pattern discovery through implicit preference and performance learning," in *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics, IDEA 2013*, Chicago, Illinois: ACM, 2013, pp. 27–35, ISBN: 978-1-4503-2329-1.

[110] M. Van Leeuwen, "Interactive data exploration using pattern mining," in *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*, Springer, 2014, pp. 169–182.

[111] B. Goethals, S. Moens, and J. Vreeken, "Mime: A framework for interactive visual pattern mining," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2011*, San Diego, California, USA: ACM, 2011, pp. 757–760, ISBN: 978-1-4503-0813-7.

[112] M. J. Zaki and B. Phoophakdee, "MIRAGE: A framework for mining, exploring and visualizing minimal association rules," Computer Science Department, Rensselaer Polytechnic Institute, Tech. Rep. 03-4, 2003.

[113] G. Liu, A. Suchitra, H. Zhang, M. Feng, S.-K. Ng, and L. Wong, "AssocExplorer: an association rule visualization system for exploratory data analysis.," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2012*, ACM, 2012, pp. 1536–1539.

[114] J. Blanchard, F. Guillet, and H. Briand, "A user-driven and quality-oriented visualization for mining association rules," in *Proceedings of the 3rd IEEE International Conference on Data Mining ICDM, Melbourne, Florida, USA*, 2003, pp. 493–496.

[115] A. Appice and P. Buono, "Analyzing multi-level spatial association rules through a graph-based visualization.," in *Proceedings of International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2005*, ser. Lecture Notes in Computer Science, vol. 3533, Springer, 2005, pp. 448–458.

[116] H. Michael, C. Sudheer, H. Kurt, and B. Christian, "The Arules R-Package Ecosystem: Analyzing Interesting Patterns from Large Transaction Data Sets," *Journal of Machine Learning Research*, vol. 12, pp. 2021–2025, 2011.

[117] S. Chakravarthy and H. Zhang, "Visualization of Association Rules over Relational DBMSs," in *Proceedings of the 2003 ACM Symposium on Applied Computing, SAC 2003*, Melbourne, Florida: ACM, 2003, pp. 922–926.

[118] G. Andrienko and N. Andrienko, "Interactive maps for visual data exploration," *International Journal of Geographical Information Science*, vol. 13, no. 4, pp. 355–374, 1999.

[119] W. Castillo-Rojas and C. Peralta Alexis and Meneses, "Augmented visualization of association rules for data mining," in *8th Alberto Mendelzon Workshop on Foundations of Data Management, AMW 2014*, Cartagena de Indias, Colombia, 2014.

[120] T. Kohonen, R. M. Schroeder, and T. S.-T. Huang, Eds., *Self-Organizing Maps*, third edition. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2001, ISBN: 3540679219.

[121] D. H. Chau, A. Kittur, J. I. Hong, and C. Faloutsos, "Apolo: Making sense of large network data by combining rich user interaction and machine learning," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2011*, Vancouver, BC, Canada: ACM, 2011, pp. 167–176, ISBN: 978-1-4503-0228-9.

[122] D. H. Chau, L. Akoglu, J. Vreeken, H. Tong, and C. Faloutsos, "TourViz: Interactive visualization of connection pathways in large graphs," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2012*, Beijing, China: ACM, 2012, pp. 1516–1519.

[123] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Exploring artificial intelligence in the new millennium," in, G. Lakemeyer and B. Nebel, Eds., San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, ch. Understanding Belief Propagation and Its Generalizations, pp. 239–269, ISBN: 1-55860-811-7.

[124] E. Galbrun and P. Miettinen, "Mining redescriptions with Siren," *ACM Transactions on Knowledge Discovery from Data, In Press*, 2016.

[125] J. Kranjc, V. Podpečan, and N. Lavrač, "Clowdflows: A cloud based scientific workflow platform.," in *ECML/PKDD (2)*, P. A. Flach, T. D. Bie, and N. Cristianini, Eds., ser. Lecture Notes in Computer Science, vol. 7524, Springer, 2012, pp. 816–819, ISBN: 978-3-642-33485-6.

[126] P. Good, *Permutation tests: a practical guide to resampling methods for testing hypotheses.* Springer Science & Business Media, 2013.

[127] Y. HOCHBERG, "A sharper bonferroni procedure for multiple tests of significance," *Biometrika*, vol. 75, no. 4, pp. 800–802, 1988.

[128] Y. Benjamini and Y. Hochberg, *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995.

Bibliography

Publications Related to the Thesis

Journal articles

M. Mihelčić, S. Džeroski, N. Lavrač, and T. Šmuc, “A framework for redescription set construction,” *Expert Systems with Applications*, vol. 68, pp. 196–215, 2017, Journal Impact Factor 16/17: 3.928.

M. Mihelčić, S. Džeroski, N. Lavrač, and T. Šmuc, “Redescription mining augmented with random forest of multi-target predictive clustering trees,” *Journal of Intelligent Information Systems*, pp. 1–34, 2017, In press, Journal Impact Factor 16/17: 1.294.

M. Mihelčić, G. Šimić, M. Babić Leko, N. Lavrač, S. Džeroski, T. Šmuc, and for the Alzheimer’s Disease Neuroimaging Initiative, “Using redescription mining to relate clinical and biological characteristics of cognitively impaired and alzheimer’s disease patients,” *PLOS ONE*, vol. 12, no. 10, pp. 1–35, 2017, Journal Impact Factor 16/17: 2.806.

Conference and workshop papers

M. Mihelčić, S. Džeroski, N. Lavrač, and T. Šmuc, “Redescription mining with multi-target predictive clustering trees,” in *Proceedings of the 4th International Workshop, New Frontiers in Mining Complex Patterns, NFMCP 2015, Held in conjunction with ECML-PKDD 2015, Porto, Portugal, September 7, 2015, Revised Selected Papers*, M. Ceci, C. Loglisci, G. Manco, E. Masciari, and Z. W. Ras, Eds. Cham: Springer International Publishing, 2016, pp. 125–143.

M. Mihelčić and T. Šmuc, “InterSet: Interactive redescription set exploration,” in *Proceedings of Discovery Science: 19th International Conference, DS 2016, Bari, Italy, October 19–21, 2016*, T. Calders, M. Ceci, and D. Malerba, Eds. Cham: Springer International Publishing, 2016, pp. 35–50.

Other Publications

Journal articles

S. Blom, M. Huisman, and M. Mihelčić, “Specification and verification of GPGPU programs,” *Sci. Comput. Program.*, vol. 95, pp. 376–388, 2014.

M. Mihelčić and M. Bohanec, “Approximating incompletely defined utility functions of qualitative multi-criteria modeling method DEX,” *Central European Journal of Operations Research (CEJOR)*, vol. 25, no. 3, pp. 627–649, 2017.

Conference and workshop papers

M. Mihelčić, N. Antulov-Fantulin, M. Bošnjak, and T. Šmuc, “Extending RapidMiner with recommender systems algorithms,” in *RapidMiner Community Meeting and Conference, RCOMM 2012*, 2012.

M. Mihelčić and M. Bohanec, “Approximating DEX utility functions with methods UTA and ACUTA,” in *Proceedings of 17th International Conference Information Society, IS 2014, Ljubljana, Slovenia*, 2014, pp. 62–65.

D. Gamberger, M. Mihelčić, and N. Lavrač, “Multilayer clustering: A discovery experiment on country level trading data,” in *Proceedings of Discovery Science - 17th International Conference, DS 2014, Bled, Slovenia, October 8-10, 2014.*, 2014, pp. 87–98.

M. Mihelčić and M. Bohanec, “Empirical comparison of three methods for approximating DEX utility functions,” in *Proceedings of the 13th International Symposium on Operational Research in Slovenia, SOR 2015. Slovenian Society Informatika, Section for Operational Research, Ljubljana*, 2015, pp. 29–34.

Other

M. Mihelčić and T. Lolić, “Satisfiability problem (SAT),” *Croatian Mathematical Electronic Journal, math.e*, vol. 21, pp. 1–16, 2012.

Biography

Matej Mihelčić was born on 15 March 1988 in Zagreb, Croatia. He attended the elementary, music elementary and high school in Delnice, and the music high school in Rijeka. He participated in various competitions in chess, mathematics, informatics and logic.

He started his studies in 2006 at the Department of Mathematics, Faculty of Science at the University of Zagreb where he obtained a Bachelor's degree in Mathematics in 2009 (weighted GPA 4.044/5.0) and a Master's degree in Computer Science and Mathematics in 2011 (GPA 4.83/5.0). The topic of his thesis was *Davis-Putnam algorithm*, performed under the supervision of Asst. Prof. Mladen Vuković. He was awarded distinction *Summa cum laude* for the overall success in the Master's studies. In the same year, he obtained the University of Zagreb Rector's award for the work *A new Parallel Heuristic for Solving Satisfiability Problem* (performed under the supervision of Asst. Prof. Goranka Nogo) and a reward for academic success in year 2011 from the Department of Mathematics, Faculty of Science, University of Zagreb.

After graduation, he worked for 6 months as a student intern at the Department of Electronics, Ruđer Bošković Institute in Zagreb, under the supervision of Asst. Prof. Tomislav Šmuc. The focus of his work was on recommender systems. He worked for one year at the University of Twente in Enschede, the Netherlands, in the field of Formal Methods (verification of OpenCL programs). His supervisor was Prof. Dr. Marieke Huisman.

Currently, he is a PhD student at the Jožef Stefan International Postgraduate School in Ljubljana, Slovenia, under the supervision of Asst. Prof. Tomislav Šmuc and Prof. Dr. Nada Lavrač. He is employed as Research Assistant at the Department of Electronics at Ruđer Bošković Institute in Zagreb. His research consists of two interleaving research directions. Development of new data mining algorithms, measures, knowledge discovery techniques and tools with the main goals of obtaining knowledge that was previously hard to obtain, to offer new ways of analysing data and to reduce the required time to perform the analyses, with current focus on redescription mining. The second research direction includes applying various methods and techniques developed in data mining or machine learning or constructing new domain-specific techniques to provide assistance in the analyses of a given domain specific research problem. He is collaborating with the domain experts on several problems in biology and has worked in the medical domain on the analysis of data from various subjects tested for dementia and Alzheimer's disease.

He also works as a part-time Teaching Assistant at the Department of Mathematics, Faculty of Science of University of Zagreb. Until now, he was a teaching assistant on 6 different programming and computer science courses.

He participated in summer school on parallel programming held in Amsterdam, in 2012, summer school on mining big and complex data held in Ohrid, in 2016 and summer schools on data science held in Split, in 2016 and 2017.

He worked on the European FP7 project CARP (project number 287767) and collaborated on the European FP7 projects e-LICO (project number 231519) and MULTIPLEX (project number 317532).