DESCRIPTIVE DATA MINING FOR PARKINSON'S DISEASE DATA ANALYSIS

Anita Valmarska

Doctoral Dissertation
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Supervisor: Prof. Dr. Marko Robnik-Šikonja, Faculty of Computer and Information Science, University of Ljubljana, Slovenia
Co-Supervisor: Prof. Dr. Nada Lavrač, Jožef Stefan Institute, Ljubljana, Slovenia

Evaluation Board:
Prof. Dr. Marko Bohanec, Chair, Jožef Stefan Institute, Ljubljana, Slovenia
Prof. Dr. Zoran Bosnić, Member, Faculty of Computer and Information Science, University of Ljubljana, Slovenia
Prof. Dr. Tomislav Šmuc, Member, Ruđer Bošković Institute, Zagreb, Croatia

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL

Anita Valmarska

DESCRIPTIVE DATA MINING FOR PARKINSON'S DISEASE DATA ANALYSIS

Doctoral Dissertation

OPISNO PODATKOVNO RUDARJENJE ZA ANALIZO PODATKOV O PARKINSONOVI BOLEZNI

Doktorska disertacija

Supervisor: Prof. Dr. Marko Robnik-Šikonja

Co-Supervisor: Prof. Dr. Nada Lavrač

Ljubljana, Slovenia, April 2018

Посветено на моите родители.

Acknowledgments

My gratitude goes to my supervisor Marko Robnik-Šikonja and co-supervisor Nada Lavrač for all their knowledge, patience, research ideas, support, and energy. I want to thank my co-supervisor Nada Lavrač for giving me the opportunity to start my academic path in the Department of Knowledge Technologies and for supporting my research ideas. Additionally, I would like to thank the members of the evaluation board: Marko Bohanec, Zoran Bosnić, and Tomislav Šmuc for the careful reading and constructive feedback.

This research would not have been possible without the financial support of the Slovenian Research Agency. I wish to thank the Department of Knowledge Technologies at the Jožef Stefan Institute, the Jožef Stefan International Postgraduate School, and the European Commision for funding the research projects HBP and PD_manager, and for supporting my research within.

I am grateful to all the people I worked with during this period. I want to thank Johannes Fürnkranz for his comments on our new algorithms for subgroup discovery and classification rule learning and Julius Stecher for the interesting discussions on inverted heuristics. My gratitude goes to Dragana Miljković and my colleagues in the PD_manager research project for interesting discussions that sprung the methodologies for analysis of Parkinson's disease progression. I also wish to thank Dimitris Gatsios and Spiros Konitsiotis for their contributions in evaluating the generated models for Parkinson's disease data analysis. My thanks also go to my work colleagues at the Department of Knowledge Technologies for providing an encouraging and friendly working environment.

Finally, I want to thank my family for the emotional support and encouragement through the years of my education. I want to thank my mother Jovanka for always being my role model and my father Živko for supporting my dreams. I wish to thank my brother Dimče, for being my friend and my ally. Last but not least, I want to thank my dearest Darko for his love, support, understanding, being my rock, and keeping me sane when things seemed to be spinning out of control.

Abstract

Parkinson's disease is the second most common neurodegenerative disorder that affects people worldwide. Its symptoms affect different aspects of patients' lives that influence the quality of life and that of their families. Given the wide variability of symptoms among different patients and also for the same patient at different times, it is difficult for the clinicians to manage the Parkinson's disease. Using data mining for analyzing of Parkinson's disease data can lead to the identification of similar patients, with the aim to assist the clinicians to respond more promptly and in a more personalized fashion to the changes of the patients' status.

The management of Parkinson's disease to improve the patients' quality of life is not widely researched. In this thesis, we address the question of data mining support for the analysis of Parkinson's disease progression and management with antiparkinson medications. The thesis is divided into three parts addressing improved descriptive modeling, grouping of patients based on similar disease progression, and analysis of the most important symptoms and medication changes.

High-quality descriptive models are highly appreciated by the clinicians. Classification rules and rules describing interesting subgroups are important components of descriptive data mining. Rule learning algorithms typically proceed in two phases: rule refinement selects the conditions for specializing the rule and rule selection selects the final rule among several rule candidates. While most conventional algorithms use the same heuristic for guiding both phases, recent research indicates that using two separate heuristics is conceptually better justified, improves the coverage of positive examples, and may result in better predictive accuracy. The thesis presents and evaluates two new beam search rule learning algorithms: DoubleBeam-SD for subgroup discovery and DoubleBeam-RL for classification rule learning. The algorithms use two separate beams and can combine various heuristics for rule refinement and rule selection, which widens the search space and allows for finding rules with improved quality. In the classification rule learning setting, the experimental results confirm previously shown benefits of using two separate heuristics for rule refinement and rule selection. In subgroup discovery, DoubleBeam-SD algorithm variants outperform several related state-of-the-art algorithms. We use the newly introduced algorithms in Parkinson's disease data analysis.

Careful management of the patient's status is crucial to ensure the patient's independence and quality of life. This is achieved by personalized treatment based on individual patient's symptoms and medical history. We address the issue of determining patient groups with similar disease progression patterns coupled with patterns of medication changes that lead to the improvement or decline of the patients' quality of life symptoms. To this end, we propose a new methodology for clustering short time series of patients' symptoms and prescribed medications. The methodology also employs time sequence data analysis using skip-grams to monitor disease progression. The results demonstrate that the motor and autonomic symptoms are the most informative for evaluating the quality of life of Parkinson's disease patients. We show that Parkinson's disease patients can be divided

into clusters ordered in accordance to the severity of their symptoms. By following the evolution of symptoms for each patient separately, we were able to determine the patterns of medication changes that lead to the improvements or degradations of the patients' quality of life.

Based on discovered groups of similar patients, we present a step towards personalized management of Parkinson's disease patients. We present two novel approaches. The first algorithm analyzes the symptoms' impact on the Parkinson's disease progression. Experiments on the Parkinson Progression Markers Initiative (PPMI) data reveal a subset of symptoms influencing the disease progression which are already established in Parkinson's disease literature, as well as symptoms that have been considered only recently as possible indicators of disease progression. The second novelty is a methodology for detecting patterns of medications dosage changes based on the patient's status. The methodology combines multitask learning using predictive clustering trees (PCTs) and short time series analysis to better understand when a change in medications is required. The experiments on PPMI data demonstrate that using the proposed methodology, we can identify some clinically confirmed patients' symptoms suggesting medications change. In terms of predictive performance, the PCT approach is comparable to the random forest model but is advantageous due to the model interpretability.

Povzetek

Parkinsonova bolezen je druga najpogostejša nevrodegenerativna motnja na svetu. Simptomi vplivajo na različne vidike življenja pacientov, kar vpliva tudi na njihovo kakovost in na kakovost življenja njihovih družin. Med simptomi, ki jih čutijo bolniki s Parkinsonovo boleznijo, obstajajo velike razlike, kar velja celo za simptome istega bolnika v različnih obdobjih. Uporaba podatkovnega rudarjenja za analizo podatkov o Parkinsonovi bolezni lahko prispeva k identifikaciji podobnih bolnikov in uspešnejši terapiji in na ta način k bolj individualnemu zdravljenju, kar bo lahko pomagalo zdravnikom, da se bodo odzivali na stanje bolnikov z bolj personaliziranimi terapijami.

Zdravljenje Parkinsonove bolezni z namenom izboljšanja kakovosti življenja bolnikov ni dobro raziskana tema. V disertaciji se ukvarjamo z rudarjenjem podatkov za podporo pri analizi in nadzoru Parkinsonove bolezni z antiparkinsonskimi zdravili. Disertacija je razdeljena na tri dele, ki obravnavajo izboljšave opisnih modelov, združevanje bolnikov na podlagi podobnega poteka bolezni ter analizo najpomembnejših simptomov in sprememb zdravil.

Zdravniki cenijo kakovostne opisne modele, ki lahko razkrijejo zanimive in nepričakovane vzorce v podatkih. Pomembne komponente opisnega podatkovnega rudarjenja so pravila za razvrščanje in pravila za opisovanje zanimivih podskupin. Učenje pravil običajno poteka v dveh fazah: faza dopolnjevanja izbere pogoje za specializacijo pravila, faza izbora pravil pa med kandidati izbere končno pravilo. Medtem ko večina algoritmov uporablja isto hevristiko v obeh fazah, nedavne raziskave kažejo, da je uporaba dveh ločenih hevristik konceptualno boljša, saj izboljša pokritost pozitivnih primerov in lahko privede do večje napovedne točnosti. V nalogi predstavimo in ovrednotimo dva nova učna algoritma: DoubleBeam-SD za odkrivanje podskupin in DoubleBeam-RL za učenje klasifikacijskih pravil. Algoritma uporabljata dva ločena preiskovalna snopa in lahko uporabita različne hevristike za dopolnjevanje in izbiro pravil, s čimer se razširi iskalni prostor in omogoči iskanje kakovostnejših pravil. Eksperimentalni rezultati potrjujejo prednosti uporabe dveh ločenih hevristik pri izgradnji klasifikacijskih pravil. Pri odkrivanju podskupin je nekaj različic algoritma DoubleBeam-SD učinkovitejših od najuspešnejših sorodnih algoritmov. Razvite algoritme uporabimo tudi pri analizi podatkov o Parkinsonovi bolezni.

Pri zdravljenju Parkinsonove bolezni je nadzor bolnikovega stanja ključnega pomena za neodvisnost in kakovost bolnikovega življenja. To dosežemo z bolniku prilagojenim zdravljenjem, ki temelji na bolnikovih simptomih in anamnezi. V delu obravnavamo vprašanje določanja skupin bolnikov s podobnimi vzorci poteka bolezni ter podobnimi vzorci spremembe zdravil, ki vodijo k izboljšanju ali poslabšanju pokazateljev kakovosti življenja bolnikov. V ta namen predlagamo novo metodologijo za analizo kratkih časovnih vrst bolnikovih simptomov in podatkov o predpisanih zdravilih. Metodologija za analizo časovnih vrst uporablja preskočne n-grame za spremljanje poteka bolezni. Rezultati kažejo, da so motorični in avtonomni simptomi najinformativnejši za ocenjevanje kakovosti življenja bolnikov s Parkinsonovo boleznijo. V delu pokažemo, da lahko bolnike razdelimo v skupine glede na resnost simptomov. S spremljanjem razvoja simptomov za vsakega posameznega

bolnika določimo vzorce sprememb zdravil, ki lahko privedejo do izboljšav ali poslabšanja kakovosti njihovega življenja.

Na podlagi določanja skupin podobnih bolnikov poskušamo izboljšati personalizirano zdravljenje bolnikov s Parkinsonovo boleznijo. Predstavljamo dva nova pristopa. Prvi algoritem analizira vpliv simptomov na potek Parkinsonove bolezni. Poskusi na podatkih PPMI (Parkinson Progression Markers Initiative) razkrivajo podmnožico simptomov, ki vplivajo na potek bolezni in so že obravnavani v literaturi o Parkinsonovi bolezni, pa tudi simptome, ki so jih začeli zdravniki šele nedavno obravnavati kot možne pokazatelje poteka bolezni. Druga novost je metodologija za odkrivanje vzorcev pri odmerjanju zdravil, ki temelji na bolnikovem stanju. Metodologija združuje večopravilno učenje dreves za napovedno razvrščanje (PCT) in analizo kratkih časovnih vrst za ugotavljanje, kdaj je potrebna sprememba zdravil. Eksperimenti na podatkih PPMI kažejo, da lahko z uporabo predlagane metodologije ugotovimo, da se zaradi sprememb zdravil pojavijo nekateri klinično potrjeni simptomi. V smislu napovedne točnosti se izkaže, da je večopravilni pristop z uporabo PCT metode primerljiv z naključnim gozdom, njegova prednost pa je interpretabilnost.

Contents

List of Figures

List of Tables

Abbreviations

RL	... Rule Learning
SD	... Subgroup Discovery
AUC	... Area Under Curve
WRACC	... Weighted Relative ACCuracy
PD	... Parkinson's Disease
PPMI	... Parkinson's Progression Markers Initiative
LEDD	... Levodopa Equivalent Daily Dosage
MAO-B	... MonoAmine Oxidase B
PCT	... Predictive Clustering Tree
NLP	... Natural Language Processing
MTL	... Multi Task Learning
MDS	... Movement Disorder Society
UPDRS	... Unified Parkinson's Disease Rating Scale
MoCA	... Montreal Cognitive Assessment
SCOPA-AUT	... SCales for Outcomes in PArkinson's disease–AUTonomic
QUIP	... QUestionnaire for Impulsive-compulsive disorders in Parkinson's disease
PASE	... Physical Activity Scale for the Elderly

Chapter 1

Introduction

This thesis contributes to the development of descriptive data mining approaches for Parkinson's disease data analysis. The main contributions include improving classification rule learning algorithms and subgroup discovery algorithms, methods for analyzing disease progression based on clustering of patients and skip-grams, and methods for analyzing changes in the dosage of medications using multitask learning. In this introductory chapter, we motivate the problems addressed and overview solutions to the presented problems. This is followed by the purposes of the dissertation, its goals, and scientific contributions. We conclude with a structural overview of the rest of the thesis.

1.1 Background and Motivation

Parkinson's disease is a neurodegenerative disorder affecting people worldwide. It is a disorder of the central nervous system, mostly affecting the patients' motor functions. According to Bega (2017), the motor symptoms most associated with the idiopathic Parkinson's disease are bradykinesia, rigidity, resting tremor, and postural instability. In addition to the motor symptoms, patients experience several emotional and behavioral problems, including depression, anxiety, impulsivity, and sleeping problems. Both motor and non-motor symptoms affect the patients' ability to live independently, consequently decreasing their quality of life and affecting also the lives of their families.

At the moment, there is no cure for Parkinson's disease. The disease treatment is directed towards managing the symptoms of the patients and prolonging their independence. The management of symptoms is done mostly by prescribing antiparkinson medications. A careful medication therapy is of crucial importance as the clinicians must prudently balance between controlling the symptoms and reducing the side effects of antiparkinson medications prolonged usage.

Parkinson's disease poses a high economic cost on countries. This cost can be divided into a direct cost caused by the decreased independence of the patients and indirect cost induced by the dedication of patients' families to care for their wellbeing. As of 2012, the estimated annual cost of Parkinson's disease on the economies of European Union countries is estimated at 13.9 billion EUR: a cost that will increase as the population ages (Olesen, Gustavsson, Svensson, Wittchen, & Jönsson, 2012). Efforts are made by the European Union to address this issue. A quick and efficient response to the patient's status is needed in order to control the disease, treat its symptoms, and consequently lower the burden on the economies of countries.

The introduction of data mining techniques can lead to a faster discovery of groups of patients with similar symptoms, therapies, and patterns of disease progression. It can reveal the common characteristics of patient groups and identify how symptoms should

be treated with medications. Parkinson's disease clinicians only have access to the data of their patients. Data mining can take advantage of databases of numerous Parkinson's disease patients from around the world to find relevant patterns in the data.

Most of machine learning and data mining research in the field of Parkinson's disease is concerned with the prediction of Parkinson's disease diagnosis (Gil & Johnson, 2009; Ramani & Sivagami, 2011), detection of patients' symptoms from wearable sensors (Patel et al., 2009), or determining subtypes of Parkinson's disease patients (Lewis et al., 2005; Ma, Chan, Gu, Li, & Feng, 2015; Reijnders, Ehrt, Lousberg, Aarsland, & Leentjens, 2009). Currently, there are no methods to follow Parkinson's disease progression or analysis of patients' medications therapy.

The usage of data mining for the analysis of Parkinson's disease patients' symptoms and therapies is further facilitated by the increased availability of real Parkinson's disease patients data. For example, such data has been available by the PD_manager: m-Health platform for Parkinson's disease management (2015), an EU Horizon 2020 project, aimed at the development of a patient-centric platform for Parkinson's disease. Providing long-term access to patients' motor and non-motor symptoms data should enable the clinicians to prescribe the best therapy for symptoms maintenance (Gatsios, Rigas, Miljkovic, Seljak, & Bohanec, 2016) and maintain a good quality of life of the patients. One of the key elements of this project is the use of machine learning for the development of a decision support system, helping the clinicians in choosing appropriate patients' therapies (Gatsios et al., 2016).

In this research, which was a part of the PD_manager project (PD_manager: m-Health platform for Parkinson's disease management, 2015), we use symptoms and medications data from the Parkinson's Progression Markers Initiative (PPMI) data collection (Marek et al., 2011). This is an extensive data collection of Parkinson's disease patients monitored through a period of five years. The PPMI data collection gives an opportunity to follow the disease progression among different groups of PD patients and to monitor how clinicians react to their symptoms–the ones resulting from the disease as well as those that are side effects of the prescribed medications.

The relatively unexplored field of data mining for Parkinson's disease data analysis offers many possibilities for data mining research. Below we present our motivation for developing novel methods in descriptive learning, ordering of groups of patients according to their symptoms severity, analysis of disease progression, and medications change analysis.

1.1.1 Classification Rule Learning and Subgroup Discovery

Good descriptive methods are essential in order to identify patients and describe them with their common symptoms. Subgroup discovery is a technique for identifying subgroups of similar patients (Gamberger & Lavrač, 2002). The task of subgroup discovery is to find interesting subgroups in the population, i.e. subgroups that have a significantly different class distribution than the entire population (Klösgen, 1996; Wrobel, 1997). The result of subgroup discovery is a set of individual rules where the rule consequence is a target class label (Fürnkranz, Gamberger, & Lavrač, 2012). Rule learning is a symbolic data analysis technique that can be used to construct understandable models or patterns describing the data (Clark & Niblett, 1989; Fürnkranz et al., 2012; Michalski, 1969). The main difference between classification rule learning and subgroup discovery is that subgroup discovery algorithms construct individual rules describing the properties of individual groups of target class instances, while classification rule learning algorithms construct a set of classification rules covering the entire problem space. The learning process in both classification rule learning and subgroup discovery typically proceeds in two phases: rule refinement selects conditions for specializing the rule, and rule selection selects the final rule among several

rule candidates.

Similarly to classification rule learning algorithms, subgroup discovery algorithms use pre-selected heuristics in their rule learning process (Fürnkranz et al., 2012). In the process of building a new rule, they do not differentiate between the selection and refinement step. Stecher, Janssen, and Fürnkranz (2014) proposed to use separate heuristics for each of the two rule construction phases, and suggested that in the refinement phase, the so-called inverted heuristics shall be used for evaluating the relative gain obtained by refining the current rule. The key idea of these heuristics is the following: while the majority of conventional rule learning heuristics, such as the Laplace or m-estimate, anchor their evaluation on the empty rule that covers all examples, inverted heuristics anchor the point of view on the base (parent) rule, which is more appropriate for a top-down refinement process (Stecher et al., 2014). As a side effect to improved rule quality in terms of classification accuracy, the rules generated by using inverted heuristics in the refinement phase are longer and contain more terms (descriptors), thus offering an additional explanation. We extend the approach of Stecher et al. (2014) to improve classification rule learning as well as subgroup discovery with two beams, taking both phases into account.

The division of patients into groups of similar symptoms can be done using method such as clustering. In the medical literature on Parkinson's disease, Goetz et al. (2015) suggested that sums of the severity of certain symptoms can be used to determine the overall status of the patient. For this reason, we explore groups of patients on data, described by the sums of the considered symptoms. Since patients' symptoms data are collected from different sources, there is a possibility to use both multi-view (Xu, Tao, & Xu, 2013) and single view methods. Multi-view clustering can use the information from multiple sources (views) to provide an alternative analysis of patients (He, Kan, Xie, & Chen, 2014). This approach has not yet been investigated in the context of Parkinson's disease.

We hypothesize that the status of patients can be learned from the descriptions of obtained clusters. These descriptions shall enable the experts to establish an ordering or a partial ordering of clusters according to the severity of the described symptoms. The ordering of clusters is important for distinguishing changes—improvement or degradation—of the patients' overall status. The clusters' descriptions are obtained by the abovementioned descriptive methods.

1.1.2 Discovering Parkinson's Disease Progression Patterns

The Parkinson's disease patients' symptoms and status data are recorded in the database and updated at regular time intervals (on each visit to the clinician). The status of patients is changing through time, reflecting a natural progression of the disease and medication intake. We expect that–after initial clustering of patients in terms of the severity of their symptoms–the patients will have changed clusters between the considered time points (between two visits to the clinician), reflecting the improvement or worsening of their overall (quality of life) status. By considering these cluster changes across all patients throughout their involvement in the study, we should be able to determine the patterns of patients symptoms changes.

The scarcity of recorded events (6 visits at most) prevents the use of traditional time series approaches. Therefore, we address the problem of disease progression with approaches adapted from natural language processing and use sequence analysis to determine patterns of disease progression. In order to increase the robustness of our results, we model the sequences of changes between clusters using skip-grams (Guthrie, Allison, Liu, Guthrie, & Wilks, 2006) instead of n-grams that are regularly used in the analysis of data sequences. The introduction of skip-grams results in an increased number of investigated n-grams,

providing more stable and robust patterns of cluster changes.

The use of skip-grams has been up to now limited to natural language processing tasks and to the best of our knowledge, except our work, no other research has yet addressed the application of skip-grams in disease progression data analysis.

1.1.3 Tracking Medication Changes

Parkinson's disease patients are treated with combinations of antiparkinsonian medications to improve the patients' quality of life and reduce the unwanted side effects. The assigned combinations depend on the patients' symptoms (Fox et al., 2011; Seppi et al., 2011), consisting of disease symptoms and side effects symptoms. The identification of interactions between the antiparkinsonian medications selected for the given symptoms can assist the clinicians when considering changing the patients' therapies. To the best of our knowledge, there are no available data mining approaches supporting the control of disease symptoms with the changes in the medications therapy. Zhao, Papapetrou, Asker, and Boström (2017) use the analysis of heterogeneous temporal data in electronic health records to detect adverse drug events. They use the history of patients symptoms in order to predict a single event (adverse drug event: yes or no), while we follow the patients' disease development and predict changes in the therapies as a result of changes in the overall status.

Patients' symptoms are collected from multiple sources. The multitask problem of simultaneously learning the dosage change of several antiparkinsonian medications can thus be addressed in two settings: using features from a single data set and features from multiple data sets. The latter setting is a representative of multitask multi-view learning and is not addressed in this thesis. The analysis of dosage changes of antiparkinsonian medications as a reaction to patients' symptoms from multiple views can lead to the identification of groups of symptoms that require specific therapy modifications. This is a step towards more personalized assistance to clinicians in handling therapies of their patients.

1.2 Purpose of the Thesis

The purpose of this dissertation is to develop methods supporting the analysis of longitudinal Parkinson's disease patients data, consisting of the symptoms data and the medications data. The dissertation covers three main topics: development of improved algorithms for classification rule learning and subgroup discovery, development of a methodology for finding patterns of Parkinson's disease progression, and development of a methodology for detection of medications dosage changes.

Development of improved algorithms for classification rule learning and subgroups discovery. The first part of the dissertation (covered in Section 4) examines algorithms for rule learning and subgroup discovery motivated by separating the two phases of the rule learning process, rule refinement and rule selection. Stecher et al. (2014) introduced the idea that due to the different nature of the refinement and selection phase of the rule learning process, it is beneficial to separate these two phases by using heuristics that take full advantage of each phase. However, they only store a single rule with the best potential to refine and the selection phase is therefore limited to a single rule. Their experimental work is focused on the separate-and-conquer algorithm for classification rule learning. Although the results are encouraging, their search strategy can miss some rules with an even better selection quality. For that purpose, we suggest keeping track of a subset of rules that have the best potential for refinement and selection, and store them separately. In this way, we expand the search space and possibly discover rules that are not found by other state-of-the-art

algorithms for classification rule learning and subgroup discovery. In our evaluation, we determine the default parameters for the introduced algorithms and compare them with other state-of-the-art algorithms.

Development of a methodology for finding patterns of Parkinson's disease progression. The current application of data mining to Parkinson's disease data is limited to diagnosis of new patients (Gil & Johnson, 2009), detection of symptoms (Timmer, Gantert, Deuschl, & Honerkamp, 1993), detection of subtypes of Parkinson's disease patients (Lewis et al., 2005), and assessing the success of deep brain stimulation surgery as a last resort in the treatment of Parkinson's disease patients (Y. Liu et al., 2014). The Parkinson's disease progession in terms of patients' motor and overall UPDRS (Unified Parkinson's Disease Rating Scale) score was addressed by Eskidere, Ertaş, and Hanilçi (2012), Tsanas (2012), Tsanas, Little, McSharry, and Ramig (2010), Tsanas, Little, McSharry, and Ramig (2010). Their evaluation of Parkinson's disease progression is done on data from non-invasive speech tests for a six months period. During this six months period all of the patients were off there antiparkinson medications. To the best of our knowledge, no data mining research is done on disease progression, reasons for progression, and clinicians' reaction to symptoms changes with modification of medications therapies of patients. This is of crucial importance in order to maintain a good quality of life for Parkinson's disease patients. We address this issue by i) developing a method that combines clustering of patients into groups of patients with similar symptoms, and ii) following changes with respect to patient's symptoms and iii) their prescribed medications therapies as the patients change clusters through time. The division of patients into clusters can be done using traditional clustering methods, e.g., k-means clustering, or multi-view clustering approaches that take advantage of groups of recorded symptoms. To determine robust patterns of disease progression, we adapt the skip-gram approach from natural language processing. With our approach, we preserve the sequential nature of the cluster assignments that reflect changes in the patients' overall status.

Development of a methodology for the detection of medications change patterns. Based on the patients' assignment to clusters between consecutive visits and the (partial) order established between the clusters, the changes in the patients' overall status can be characterized as positive or negative. Our methodology from the previous point is able to detect patterns of medications dosage changes when the overall status of the patients has improved or deteriorated. However, this approach does not address the underlying symptoms that affected the changes and which caused the clinicians to change the medications dosages. For this purpose, we propose multi-task learning with predictive clustering trees (PCT) to determine when the clinicians decide to change the patients' therapies and what are the changes. Based on this knowledge extracted from the real patients' data, we consulted the experts concerning the discovered scenarios which enabled us to develop of decision support models, which should enable more personalized disease management of Parkinson's disease patients.

Public accessibility of the developed algorithms. Our purpose is also to make the algorithms for classification rule learning and subgroup discovery publicly accessible. The code for the DoubleBeam-RL algorithm and the DoubleBeam-SD algorithm is publicly available on GitHub. We are currently working on its implementation within the ClowdFlows platform (Kranjc, Podpečan, & Lavrač, 2012), i.e. an open-source, cloud-based platform for composition, execution, and sharing of interactive machine learning and data mining workflows. On the other hand, the code we developed for

the analysis of Parkinson's disease data is available upon request as is closely related to the data being analyzed. We do not have permission to share the data but users can obtain it from PPMI.

1.3 Goals and Hypotheses of the Thesis

The goals and hypotheses of this thesis are aligned with its purposes described in the previous section. We believe that by separating the two phases of the rule learning process, the classification rule learning algorithms and the subgroup discovery algorithms will generate rules with the improved quality compared to the rules produced by their state-of-the-art counterparts. The introduction of two beams and separate heuristics for each of the phases in the rule learning process will widen the search space and enable the algorithms to construct rules which would be missed by the standard rule learning algorithms.

We hypothesize that Parkinson's disease patients can be divided into groups of patients with similar symptoms. These groups of patients can be partially ordered[1] according to the severity of the symptoms describing each of the clusters. The patients' status will change through time, thus prompting the patients to be assigned to different groups. Given the fact that the clusters are at least partially ordered in terms of the symptoms severity, the transition between two clusters can also indicate a transition in the overall status of the patient, signaling an improvement or deterioration of the patients' status. Following the changes of clusters for all patients using skip-grams, our methods will reveal robust patterns of disease progression, while further analysis of these patterns will reveal characteristics of the patients following certain patterns of disease progression.

Analysis of medications dosage changes aligned with the changes of patients' status can reveal patterns that led to the improvement or degradation of the overall status of patients. We hypothesize that using multitask approach, we will be able to determine the symptoms that trigger the changes in medication dosages.

1.4 Scientific Contributions

The scientific contributions of the thesis are as follows.

Contribution 1 We developed improved algorithms for classification rule learning and subgroup discovery by separating the two phases of the rule learning process, rule refinement and rule selection. We used two separate beams for the refinement and selection phase and used different heuristics in each of the phases. We evaluated and compared the newly developed algorithms to their state-of-the-art counterparts. We made the code publicly available on GitHub[2].

Publications related to this contribution

Journal Paper

Valmarska, A., Lavrač, N., Fürnkranz, J., & Robnik-Šikonja, M. (2017). Refinement and selection heuristics in subgroup discovery and classification rule learning. *Expert Systems with Applications*, *81*, 147–162. doi:10.1016/j.eswa.2017.03.041

[1] In a partially ordered set of clusters, not every pair of clusters need be comparable.
[2] https://github.com/bib3rce/RL_SD

Conference Paper

Valmarska, A., Robnik-Šikonja, M., & Lavrač, N. (2015). Inverted heuristics in subgroup discovery. In *Proceedings of the 18th International Multiconference Information Society* (Vol. 178, pp. 41–44).

Contribution 2 We developed a methodology for the division of patients into clusters based on the severity of the symptoms of patients assigned to each cluster. We showed that the resulting clusters can be at least partially ordered according to the severity of the disease. Using this, we can declare the changes of clusters between two consecutive visits to be positive or negative (the patient's status has improved or has worsened). Based on these changes, we presented an algorithm for detection of medications dosage changes that occurred most frequently when the status of the patients improved or worsened. We also present a methodology for determining patterns of Parkinson's disease progression based on skip-grams from natural language processing. The pseudocode of the methodology is presented in Chapter 5 and Appendix A.1.

Publications related to this contribution

Journal Paper

Valmarska, A., Miljkovic, D., Lavrač, N., & Robnik-Šikonja, M. (2018). Analysis of medications change in Parkinson's disease progression data. *Journal of Intelligent Information Systems*. doi:10.1007/s10844-018-0502-y

Conference Paper

Valmarska, A., Miljkovic, D., Robnik-Šikonja, M., & Lavrač, N. (2016). Multi-view approach to Parkinson's disease quality of life data analysis. In *Proceedings of the International Workshop on New Frontiers in Mining Complex Patterns* (pp. 163–178). Springer.

Contribution 3 We developed a methodology for the detection of medications change patterns and determining the symptoms that trigger the change of the dosages of antiparkinson medications. We found patterns of medications dosage changes as a result of changes in the overall status (quality of life) of patients. We developed an algorithm for determining the symptoms that have the strongest impact on the progression of the disease. Some of the identified symptoms are well known, while others only recently started to gain recognition as possible markers of Parkinson's disease progression. The pseudocode of the methodology is presented in Appendix A.2.

Publications related to this contribution

Journal Paper

Valmarska, A., Miljkovic, D., Konitsiotis, S., Gatsios, D., Lavrač, N., & Robnik-Šikonja, M. (2018). Symptoms and medications change patterns for Parkinson's disease patients stratification. *Artificial Intelligence in Medicine (accepted)*.

Conference Paper

Valmarska, A., Miljkovic, D., Konitsiotis, S., Gatsios, D., Lavrač, N., & Robnik-Šikonja, M. (2017). Combining multitask learning and short time series analysis in Parkinson's disease patients stratification. In *Proceedings of the Conference on Artificial Intelligence in Medicine in Europe* (pp. 116–125). Springer.

1.5 Structure of the Thesis

The remainder of the thesis is structured as follows. Chapter 2 presents the related work. Chapter 3 presents the Parkinson's disease data used in the experiments. Chapters 4, 5, and 6 present the main body of our work.

Chapter 3 is an introduction to Parkinson's disease through the description of the patients' symptoms and the medications treatments available to the clinicians to control the symptoms and promote the patients' independence. We present the PPMI data collection and describe the process of data collection, as well as the nature of symptoms, represented by the attributes and their values. We present value ranges of the attributes and statistical associations with the improvements or degradations of the patients' status.

Chapter 4 presents our approach to improving algorithms for classification rule learning and subgroup discovery by separating the two phases of rule learning, refinement and selection. We extend the idea of Stecher et al. (2014) by using two beams that hold the best rules according to their potential quality for rule refinement and selection. This chapter is divided into two sections, where we first introduce the problem description and then present the paper published in the *Expert Systems with Applications* journal that addresses the described problem.

In Chapter 5 we present our methodology for separating the patients into groups based on the similarity of their symptoms and determining patterns of medications change and disease progression. We present an algorithm for detection of medications changes patterns that most frequently cause improvement or worsening of patients' status. We employ skip-grams from natural language processing to obtain the patterns of disease progression. The chapter is divided into two sections, where we first introduce the problem description and then we present the paper published in the *Journal of Intelligent Information Systems* that addresses the described problem.

Chapter 6 presents our methodology for the detection of medications changes patterns based on the patients' status. We employ predictive clustering trees in order to learn symptoms that trigger the medications dosage change. We present an algorithm for determining symptoms that mostly influence the Parkinson's disease progression, i.e. symptoms that improve or worsen most frequently as the status of the patients improves or declines. Similarly to Chapters 4 and 5, this chapter is also divided into two sections: problem description and the paper published in the *Artificial Intelligence in Medicine* journal which addresses the described problem.

Chapter 7 summarizes the work and presents the ideas for future improvements.

Chapter 2

Related Work

This chapter provides the related work in three research topics covered by the thesis: rule learning and subgroup discovery algorithms based on separation of refinement and selection phase in rule learning, Parkinson's disease data clustering and skip-gram analysis of short series, and multi-target approach to Parkinson's disease management.

Rule learning is a standard symbolic data analysis technique used for constructing comprehensible models and patterns. Its main advantage over the other data analysis techniques is its simplicity and comprehensibility of its outputs. Rule learning has been extensively used both in predictive and descriptive rule learning settings, whereby applying different rule evaluation heuristics different trade-offs between the consistency and coverage of constructed rules can be achieved. This chapter presents a short overview of classification rule learning in Section 2.1 and subgroup discovery in Section 2.2, followed by an overview of relevant multi-view clustering approaches (Section 2.3), a short overview of methods for short time series analysis (Section 2.4), and the introduction of skip-grams for sequence data analysis (Section 2.5). In Parkinson's disease management several groups of medications are used together. We apply multi-target modeling with predictive clustering trees to capture their joint effects. We discuss related work from multitarget and multi-task learning in Section 2.6. We are interested in the importance of symptoms affecting the overall status of the disease, which is a problem addressed in feature ranking/evaluation research. We compare and contrast the proposed algorithm with existing approaches in Section 2.7.

2.1 Classification Rule Learning

The task of classification rule learning is to find models that would ideally be *complete* (i.e., cover all the positive examples, or at least most of the positives), and *consistent* (i.e., do not cover any of the negative examples, or at most a very small number of negatives). Multi-class classification problems can be addressed by using the one-versus-all approach, which learns one rule set for each class, where the examples labeled with the chosen class are considered as positive target class examples and the examples of all the other classes as the negatives.

There are numerous classification rule learning algorithms, the most popular being AQ, CN2 and Ripper. The AQ algorithm (Michalski, 1969) was the first covering algorithm for rule set construction. It is a top-down beam search algorithm that uses a random positive example as a seed for finding the best rule. The CN2 algorithm (Clark & Niblett, 1989) combines the ideas from the AQ algorithm and the decision tree learning algorithm ID3 (Quinlan, 1983), where each path from the root of the tree to a tree leaf can be viewed as a separate rule. It constructs an ordered decision list by learning rules describing the

majority class examples. Once the learned rule is added to the decision list, all the covered examples, both positive and negative, are removed from the training data set, and the rule induction process is continued on the updated training set. Ripper (Cohen, 1995) is the first rule learning algorithm that effectively overcomes the overfitting problem and is thus a very powerful rule learning system. The algorithm constructs rule sets for each of the class values. Initially, the training data set is divided into a *growing* and a *pruning* set. Rules are learned on the growing set and then pruned on the *pruning* set by incrementally reducing the error rate on the pruning set. A pruned rule is added to the rule set if the description length of the newly constructed rule set is at most d bits longer (a parameter) than the already induced rule set. Otherwise, the rule learning process is stopped. Similarly to the CN2 algorithm, when a new rule is added to the rule set, all the instances covered by this rule are removed from the growing set. In addition to pruning the rules before adding them to the induced rule set, Ripper prevents rules overfitting in a post-processing phase in which the learned rule set is optimized and the selected rules are re-learned in the context of the other rules. FURIA (Hühn & Hüllermeier, 2009) is a classification rule learning algorithm that extends the Ripper algorithm by learning fuzzy rules.

Despite its long history, rule learning is still actively researched and routinely applied in practice. For example, Napierala and Stefanowski (2015) use rule learning with argumentation to tackle imbalanced data sets, and Ruz (2016) explores the order of instances in seeding rules to improve the classification accuracy. Minnaert, Martens, De Backer, and Baesens (2015) discuss the importance of proper rule evaluation measures for improving the accuracy of classification rule learning algorithms. They also introduce multi-criteria learning and investigate a Pareto front as a trade-off between comprehensibility and accuracy of rule learners.

In the line of research started by Parpinelli, Lopes, and Freitas (2002), rule learning is turned into an optimization problem using an ant colony optimization approach. The initial rule learning algorithm, named Ant-Miner, worked for nominal attributes only but was later improved by Pičulin and Robnik-Šikonja (2014) to efficiently handle numeric attributes. Classification rule learning has been a vivid topic of research also in inductive logic programming and relational data mining. For example, Zeng, Patel, and Page (2014) developed the QuickFOIL algorithm that improves over the original FOIL algorithm (Quinlan & Cameron-Jones, 1993).

Learning rules can be regarded as a search problem (Mitchell, 1982). Search problems are defined by the structure of the search space, a search strategy for searching through the search space, and a quality function (a *heuristic*) that evaluates the rules in order to determine whether a candidate rule is a solution or how close it is to being a solution to be added to the rule set, i.e. the final classification model. The search space of possible solutions is determined by the modeling language bias (Fürnkranz et al., 2012). In propositional rule learning, the search space consists of all the rules of the form $targetClass \leftarrow Conditions$, where $targetClass$ is one of the class labels, and $Conditions$ is a conjunction of features. Features have the form of $A_i = v_{ij}$ (attribute A_i has value v_{ij}).

For learning a single rule, most learners use one of the following search strategies: *general-to-specific* (*top-down hill-climbing*) or *specific-to-general* (*bottom-up*), where the former is more commonly used. Whenever a new rule is to be learned, the learning algorithm initializes it with the *universal rule* $\mathbf{r}^\top$. This is an empty rule that covers all the examples, both positive and negative. In the rule refinement phase, conditions are successively added to this rule, which decreases the number of examples that are covered by the rule. Candidate conditions are evaluated with the goal of increasing the consistency of the rule while maintaining its completeness, i.e. a good condition excludes many negative examples and maintains good coverage of the positive examples.

Heuristic functions are used in order to evaluate and compare different rules. Different heuristics implement different trade-offs between the two objectives (coverage and consistency). While CN2 and Ripper use entropy as the heuristic evaluation measure, numerous other heuristic functions have been proposed in rule learning—for a variety of heuristics and their properties the interested reader is referred to (Fürnkranz et al., 2012). The most frequently used heuristics in rule learning are:

Precision:

$$\mathrm{h}_{prec}(p,n) = \frac{p}{p+n} \tag{2.1}$$

Laplace:

$$\mathrm{h}_{lap}(p,n) = \frac{p+1}{p+n+2} \tag{2.2}$$

m-estimate:

$$\mathrm{h}_{m\text{-}est}(p,n,m) = \frac{p+m\cdot\frac{P}{P+N}}{p+n+m} \tag{2.3}$$

where, for a given rule, arguments p and n denote the number of positive and negative examples covered by the rule (i.e. the true and false positives, respectively), and P and N denote the total number of positive and negative examples in the data set. Given that these heuristics concern the problem of selecting the best of multiple refinements of the same base rule (the empty rule, universal rule), values P and N can be regarded as constant, so that the above functions may be written as $h(p,n)$ depending only on the true and false positives, while in Equation (2.3) the function arguments include m, which is a positive number denoting a correction towards the prior probability of the positive class (Fürnkranz et al., 2012).

Table 2.1 compares the DoubleBeam-RL classification rule learning algorithm introduced in Chapter 4, to the state-of-the-art classification rule learners that were used in the experiments. CN2 and DoubleBeam-RL are beam search algorithms, while Ripper and SC-ILL are greedy algorithms, adding conditions to the rules that maximize their respective heuristics. The DoubleBeam-RL and SC-ILL algorithms use separate heuristics adapted for the refinement and selection phase of the rule learning process. Ripper is the only considered classification rule learning algorithm that employs rule pruning and optimization of rule sets in post-processing. The algorithms use different stopping criteria; for example, Ripper uses a heuristic based on the minimum description length (MDL) principle.

Table 2.1: Comparison of the DoubleBeam-RL algorithm to the state-of-the-art classification rule learners CN2, Ripper and SC-ILL.

Algorithm	Type of search	Separate refinement heuristic	Stopping criterion	Rule pruning	Post-processing
CN2	beam	no	no beam improvement	no	no
Ripper	greedy	no	MDL	yes	yes
SC-ILL	greedy	yes	no negative examples covered	no	no
DoubleBeam-RL	beam	yes	*maxSteps*	no	no

2.2 Subgroup Discovery

The goal of data analysis is not only to build prediction models but frequently the aim is to discover individual patterns that describe regularities in the data (Fürnkranz et al., 2012; Kralj Novak, Lavrač, Zupan, & Gamberger, 2005; Wrobel, 1997). This form of data analysis is used for data exploration and is referred to as *descriptive induction*.

Subgroup discovery is a form of descriptive induction. The task of subgroup discovery is to find subgroups of examples which are sufficiently large while having a significantly different distribution of target class instances than the original target class distribution. Like in classification rule learning, individual subgroup descriptions are represented as rules in the form $targetClass \leftarrow Conditions$, where $targetClass$ is the target class representing the property of interest, and $Conditions$ is a conjunction of features that are characteristic for a selected group of individuals.

Subgroup discovery is a special case of a more general task of rule learning. Classification rule learners have been adapted to perform subgroup discovery with heuristic search techniques drawn from classification rule learning. These algorithms apply constraints, which are appropriate for descriptive rule learning. Research in the field of subgroup discovery has developed in different directions. Exhaustive methods, which include EXPLORA (Klösgen, 1996), SD-MAP (Atzmüller & Puppe, 2006), and APRIORI-SD (Kavšek, Lavrač, & Jovanoski, 2003), guarantee the optimal solution given the optimization criterion. The APRIORI-SD algorithm draws its inspiration from the association rule learning algorithm APRIORI (Agrawal & Srikant, 1994) but restricts it to constructing rules that have only the target variable (the property of interest) in their head, with *weighted relative accuracy* (WRACC), defined in Equation (2.5), used as a measure of rule quality. In order to improve the inferential power of the subgroup describing rules, the APRIORI-SD algorithm uses a post-processing step to reduce the generated rules to a relatively small number of diverse rules. This reduction is performed using the weighted covering method proposed by Gamberger and Lavrač (2000). When a rule is added to the induced rule set, weights of examples covered by the rule are decreased. This allows the method to prioritize rules which cover yet uncovered examples, thus promoting the coverage of diverse groups of examples.

While the APRIORI-SD algorithm adapts the process of association rule learning to the context of subgroup discovery, the SD subgroup discovery algorithm (Gamberger & Lavrač, 2002) performs heuristic beam search, where rule quality is estimated using the heuristic

$$\mathrm{h}_g(p, n, g) = \frac{p}{n + g}, \tag{2.4}$$

where p is the number of *true positives*, n is the *number of false positives*, and g is the *generalization parameter*. High-quality rules will cover many target class examples and a low number of non-target examples. The number of tolerated non-target examples covered by a rule is regulated by generalization parameter g. For small g, more specific rules are generated, while for bigger values of g the algorithm constructs more general rules. The interpretation of the rules produced by the SD algorithm is improved using the above mentioned weighted covering method in post-processing (Gamberger & Lavrač, 2000).

CN2-SD (Lavrač, Kavšek, Flach, & Todorovski, 2004) is a beam search algorithm, which adapts the CN2 (Clark & Niblett, 1989) classification rule learner to subgroup discovery. CN2-SD introduced a weighted covering algorithm, where examples that have already been covered by one of the learned rules are not removed from the training data set, but instead, their weights are decreased. The authors propose and compare different measures for rule evaluation. They argue that the most important measure for subgroup evaluation is *weighted relative accuracy* (WRACC), referred to as *unusualness*, defined as

Table 2.2: Some properties of subgroup discovery algorithms DoubleBeam-SD, APRIORI-SD, SD, and CN2-SD.

Algorithm	Type of search	Separate refinement heuristic	Stopping criterion	Post-processing
APRIORI-SD	exhaustive	no	minSup, minConf	yes
SD	beam	no	no beam improvement	yes
CN2-SD	beam	no	no beam improvement	no
DoubleBeam-SD	two beams	yes	*maxSteps*	optional

follows

$$\mathrm{WRACC}(p,n) = \frac{p+n}{P+N} \cdot \left(\frac{p}{p+n} - \frac{P}{P+N} \right) \tag{2.5}$$

This measure reflects both the rule significance and rule coverage, as subgroup discovery is interested in rules with significantly different class distribution than the prior class distribution that covers many instances. WRACC is the measure of choice in our experimental work on subgroup discovery for comparing the quality of the induced subgroup describing rules.

Subgroup discovery was used also in the context of semantic data mining. Adhikari, Vavpetič, Kralj, Lavrač, and Hollmén (2014) have explained mixture models by applying the semantic subgroup discovery system Hedwig (Vavpetič, Kralj Novak, Grčar, Mozetič, & Lavrač, 2013) to structure the search space and to formulate generalized hypotheses by using concepts from the given domain ontologies.

Table 2.2 compares the DoubleBeam-SD algorithm (introduced in Chapter 4) to the state-of-the-art subgroup discovery algorithms APRIORI-SD, CN2-SD, and SD, which were used in our experiments. The latter algorithms use only a single heuristic for rule evaluation, designed to optimize the selection of best rules. The DoubleBeam-SD algorithms can use pairs of different heuristics (see Chapter 4) which can be applied to estimate rule quality in both the *refinement* and *selection* phases of the rule learning process. The DoubleBeam-SD algorithm stops the learning process after a predetermined number of steps (*maxSteps*). The SD and CN2-SD algorithms stop when there are no improvements of rules in the beam, i.e. when newly induced rules have lower quality than the rules already included in the beam. APRIORI-SD uses minimal support and coverage as the stopping criteria.

2.3 Multi-View Clustering

Multi-view clustering is concerned with clustering of data by considering the information shared by each of the separate views. Many multi-view clustering algorithms initially transform the available views into one common subspace (early integration), where they perform the clustering process (Xu et al., 2013). Chaudhuri, Kakade, Livescu, and Sridharan (2009) propose a method for multi-view clustering where the translation to a lower vector space is done with Canonical Correlation Analysis (CCA). Tzortzis and Likas (2009) propose a multi-view convex mixture model that locates clusters' representatives (exemplars) using all views simultaneously. These exemplars are identified by defining a convex

mixture model distribution for each view. Cleuziou, Exbrayat, Martin, and Sublemontier (2009) present a method where in each view they obtain a specific organization using fuzzy k-means (Bezdek, 1981) and introduce a penalty term in order to reduce the disagreement between organizations in different views. Cai, Nie, and Huang (2013) propose a multi-view k-means clustering algorithm for big data. The algorithm utilizes a common cluster indicator in order to establish common patterns across the views.

Co-training (Blum & Mitchell, 1998) is one of the earliest representatives of multi-view learning. This approach considers two views consisting of both labeled and unlabeled data. Using labeled data, co-training constructs a separate classifier for each view. The most confident predictions of each classifier on the unlabeled data are then used to iteratively construct additional labeled training data. Kumar and Daumé (2011) apply the co-training principle (Blum & Mitchell, 1998) in unsupervised learning. Clustering is performed on both views, then cluster points from one view are used to modify the clustering structure of the other view. Appice and Malerba (2016) employ the co-training principle in the multi-view setting for process mining clustering. The above-mentioned approaches presume that each of the respective views is capable of producing clusters of similar quality when considered separately. He et al. (2014) do not make that presumption. They combine multiple views under a principled framework, CoNMF (Co-regularized Non-negative Matrix Factorization), which extends NMF (Non-negative Matrix Factorization) for multi-view clustering by jointly factorizing the multiple matrices through co-regularization. The matrix factorization process is constrained by maximizing the correlation between pairs of views, thus utilizing information from each of the considered views. CoNMF is a multi-view clustering approach with intermediate integration of views, where different views are fused during the clustering process. The co-regularization of each pair of views makes the clustering process more robust to noisy views. The decision to use the CoNMF approach in our work was made based on this property and on the availability of its Python code. The CoNMF approach is used in Chapter 5.

2.4 Analysis of Short Time Series

A time series is a sequence of data points indexed in time order. Time series data analysis was used to study a wide range of biological and ecological systems (Bence, 1995). The use of time series allows for studying the dynamics of a system. Short time series (8 points or less) constitute more than 80% of all time series data sets (Ernst, Nau, & Bar-Joseph, 2005). The small number of available time points does not allow for identification of statistically significant temporal profiles (Ernst & Bar-Joseph, 2006). Bence (1995) examines methods for adjusting confidence intervals of the mean and parameters of a linear regression for autocorrelation. De Alba, Mendoza, et al. (2007) suggest that simpler models can be more effective on short time series. They show that the Bayesian approach is superior to the traditional approach when applied to short time series but inferior when applied to longer time series. A lot of the research in short time series analysis is related to the analysis of short time series on microarray gene expression data. Ernst et al. (2005) present a method for clustering of short time series gene expression data, followed by the introduction of the STEM (Short Time-series Expression Miner) software program (Ernst & Bar-Joseph, 2006) specifically designed for the analysis of short time series microarray gene expression data.

In the healthcare domain, Choi, Schuetz, Stewart, and Sun (2017) incorporate temporal modeling using the recurrent neural network (RNN) model to predict heart failure. Imhoff, Bauer, Gather, and Löhlein (1998) apply short time series analysis to monitor lab variables after liver surgery and to offer support to clinicians in their decision-making process for

the treatment of acute respiratory distress syndrome. Schieb, Mobley, George, and Casper (2013) evaluate the clustering of stroke hospitalization rates, patterns of the clustering over time, and associations with community-level characteristics. They generate clusters of high and low-stroke hospitalization rates during two periods of time. According to the place of residence of patients, counties in the USA are assigned to clusters. Following the transition of counties between clusters between these two periods, counties are labeled as having a persistently high, transitional, or persistently low-stroke hospitalization rate.

Murugesan et al. (2017) present a hierarchical multi-scale approach for visualizing spatial and functional cluster evaluation patterns. Their visualization method is a two-stage method based on a sequence of community detection at each time stamp and community tracking between steps. Greene, Doyle, and Cunningham (2010) address the issue of identifying communities in dynamic networks. Appice (2017) uses social network analysis as a basic approach for organizational mining, aimed at understanding the life cycle of dynamic organizational structures.

Zhao et al. (2017) explore different representations of temporal data from electronic health records to improve prediction of adverse drug events. They obtain sequences of symbols by transforming time series of individual feature into strings, as presented in (Lin, Keogh, Wei, & Lonardi, 2007). These strings reflect the temporal nature of the original values. Results from their empirical investigation show that transformation of sequences to tabular form based on edit distance of sub-sequences to representative shapelets leads to improved predictive performance. This approach reduces the feature sequence diversity by finding informative random sub-sequences. The goal of Zhao et al. (2017) is to predict whether patients will develop adverse drug reactions. They use the history of patients symptoms in order to predict a single event (adverse drug event: yes or no), while we follow the patients' disease development and changes in their overall status as a result of therapy changes. Another difference is our use of skip-grams which reduces noise and enforces strong transition patterns.

To the best of our knowledge, the temporal nature of medical data has not been explored in research directed towards determining the progression of Parkinson's disease and determining the therapy recommendations in order to stabilize the disease progression. We present a clustering based methodology on short time series symptoms data of Parkinson's disease patients in an attempt to discover how the disease develops through time, reflected by the change of patients' symptoms. Simultaneously, we use the temporal data about their medications therapy to determine how clinicians react to patients' symptoms changes. Each Parkinson's disease patient is described with his/her symptoms and medications treatment through time. The temporal data is flattened to records from single time points, where any change of patients' symptoms between two consecutive points is referred as a change in their status. Changes in status are then connected to possible changes in medications therapies.

2.5 Skip-grams for Sequence Data Analysis

Patient's allocation to clusters in sequential time points can be viewed as a sequence of items. Analysis of contiguous sequences of items for every patient's cluster allocation can provide an insight into the disease progression and reveal patterns how (and how often) the patient's symptoms improve or degrade.

We use an approach to sequence data analysis, where we borrow the methodology initially developed in the field of natural language processing (NLP). In NLP, a contiguous sequence of n items from a given sequence of text or speech is called an n-gram (Broder, Glassman, Manasse, & Zweig, 1997). Skip-grams are a generalization of n-grams in which

the components (typically words) need not be consecutive in the text under consideration but may leave gaps that are skipped over (Guthrie et al., 2006). They provide a way of overcoming the data sparsity problem found with conventional n-gram analysis.

Another use of skip-grams is in producing word embeddings into a vector form to reduce the dimensionality and sparsity of a bag-of-words representation. Mikolov, Chen, Corrado, and Dean (2013) proposed word2vec embedding based on deep learning, which has subsequently been used in many NLP applications, including some with clinical text data (De Vine, Zuccon, Koopman, Sitbon, & Bruza, 2014) and to learn relationships between clinical processes and unified medical language system (UMLS) concepts (Choi et al., 2017). Our use of skip-grams is entirely different as we do not use embeddings but use skip-grams directly as a more robust version of n-grams.

In the context of our analysis, skip-grams allow for robust identification of frequent paths through clusters and reveal typical disease progression patterns. The patient's overall status at a given visit to the clinician, as determined by the (patient, visit) pair cluster assignment, can be seen as an item, and changes of clusters as sequences of items, which can be analyzed with the skip-grams-based approach developed for NLP. This is novel in the analysis of Parkinson's disease data and allows us to follow the progression of the patient's overall status without taking into account noise in the form of sudden changes in the patient's status. Such changes are not necessary due to Parkinson's disease but can be attributed to other stressful events in the patient's life (such as loss of a loved one, loss of a pet, etc.). The patients' status is determined by the symptoms recorded during their visit to the clinician. Patients' assignment to the clusters is determined based on their overall status. To the best of our knowledge, there has not been any study involving skip-grams that uses the actual symptoms of patients in order to explore patients' disease progression and the clinicians' response by changing the medications therapy. The skip-gram approach is used in Chapter 5.

2.6 Multitask Learning

In multitask learning (MTL), multiple related tasks are learned simultaneously on a shared attribute space. Compared to single task learning, MTL can improve model generalization and prevent overfitting (Caruana, 1997). This is achieved by transfer of intermediate knowledge (features) between jointly learned tasks.

Caruana, Baluja, and Mitchell (1996) use knowledge from the future to rank patients according to their risk to die from pneumonia. The shared attribute space consists of patients at the time they are admitted to the hospital. The multiple tasks which are learned by the model are a set of hospital tests performed to determine whether the patients are at risk of dying of pneumonia. Zhou, Liu, Narayan, Ye, and ADNI (2013) use multitask learning to model Alzheimer's disease progression. They use two clinical/cognitive measures, Mini Mental State Examination (MMSE) and Alzheimer's Disease Assessment Scale cognitive subscale (ADAS-Cog) as multiple evaluations to determine the progression of the disease. Zhang, Shen, and ADNI (2012) propose a multitask model for prediction of multiple regression and classification variables in Alzheimer's disease, which takes advantage of the multi-modal nature of patient's symptoms. Similarly to Parkinson's disease patients, Alzheimer's disease patients can be described by symptoms collected from multiple sources. All of these approaches use quantitative data "from the future" (test values) to determine how the disease progresses. Unfortunately, there are no tests to appropriately measure the progression of Parkinson's disease. None of the above-mentioned methods look at the medications that the patients are receiving to decelerate the disease progression.

We use multitask learning with the aim to simultaneously predict the values of several

target attributes (medications in our case). We use the MTL implementation with predictive clustering trees (PCTs) (Blockeel & De Raedt, 1998; Blockeel, Raedt, & Ramon, 1998). This method adapts the basic top-down induction of decision trees with clustering and allows for multitask learning. The PCT learning algorithm used is implemented in the CLUS data mining framework (Blockeel, 1998; Blockeel et al., 1998; De Raedt & Blockeel, 1997; Kocev, Vens, Struyf, & Džeroski, 2007; Piccart, Struyf, & Blockeel, 2008). We obtain multi-target decision trees, simultaneously predicting three target variables: change of levodopa dosage, change of dopamine agonists dosage, and change of MAO-B inhibitors dosage, referring to three most important medication groups used in Parkinson's disease patient management. The PCT-based approach is used in Chapter 6.

2.7 Feature Evaluation

Feature subset selection can improve the accuracy, efficiency, applicability, and comprehensibility of a learning process and its resulting model (Arauzo-Azofra, Aznarte, & Benítez, 2011; Guyon & Elisseeff, 2003; Guyon, Gunn, Nikravesh, & Zadeh, 2008). For this reason, many feature subset selection approaches have been proposed. In general, three types of feature selection methods exist: wrapper, filter, and embedded methods. Wrapper methods use the performance of a given learning algorithm as the criterion to include/exclude attributes. Embedded methods use feature selection as an integral part of their learning process. Filter methods introduce some external criterion independent of the predictor and evaluate features according to that criterion, which allows for ranking of features and selection of a suitable subset. This is fit for our purpose.

Our approach to determine the importance of symptoms for the overall disease progression is strongly related to the well-known Relief family of algorithms (Kira & Rendell, 1992; Kononenko, 1994; Reyes, Morell, & Ventura, 2015; Robnik-Šikonja & Kononenko, 2003). These algorithms evaluate attributes based on their ability to distinguish between similar instances with different class values. Contrary to the majority of feature evaluation heuristics (e.g., information gain, gini index, etc.) that assume conditional independence of attributes with respect to the target variable, the Relief approaches do not make this assumption and are suitable for problems that involve feature interaction. The Relief algorithms randomly select an instance and find the nearest instance from the same class and nearest instances from different classes. When comparing feature values of near instances the algorithm rewards features that separate instances with different class values and punishes features that separate instances with the same class value. The whole process is repeated for a large enough sample. The approach we propose also uses similar instances but uses cluster membership as a criterion for similarity instead of a distance in the feature space. When updating the importance of features, our approach assesses joint transitions from one cluster to another or from a better patient status to a worse one, while Relief algorithms use similarities in target variable.

Most feature selection methods do not attempt to uncover causal relationships between feature and target and focus instead on making best predictions. The introduction of causal feature selection (under broad assumptions) can exibit strong feature set parsimony, high predictivity, and local causal interpretability (Aliferis, Statnikov, Tsamardinos, Mani, & Koutsoukos, 2010a, 2010b). In the medical context, the Markov blanket discovery algorithm HITON (Aliferis, Tsamardinos, & Statnikov, 2003) has been used to understand physicians' decisions and their guideline compliance in the diagnosis of melanomas.

Some recent feature selection approaches try to explore the interconnection between the features by exploring the similarity graph of features (Rana et al., 2015; Shang, Wang, Stolkin, & Jiao, 2016). Other approaches pose feature selection as an optimization problem.

For example, Sun et al. (2012) use optimization in combination with a game theory based method. Our approach also uses a graph of transitions between clusters to assess similarity of patients, but we work in an unsupervised scenario and use time order of patients' visits as links between nodes. The details are explained in Chapter 6.

2.8 Parkinson's Disease Related Data Mining Research

The main branches of data mining research in the field of Parkinson's disease (PD) include: classification of PD patients, detection of subtypes of PD patients, detection of possible biomarkers, detection of PD symptoms, and assessing the success of deep brain stimulation surgery as a last resort in the treatment of Parkinson's disease patients.

The use of classification techniques offers decision support to specialists by increasing the accuracy and reliability of diagnosis and reducing possible errors. Gil and Johnson (2009) use Artificial Neural Networks (ANN) and Support Vector Machines (SVM) to distinguish PD patients from healthy subjects. Ramani and Sivagami (2011) compare the effectiveness of different data mining algorithms in the diagnosis of PD patients. Identification of Parkinson's disease subtypes is presented in the work of Lewis et al. (2005), and has been confirmed in the conclusions from Reijnders et al. (2009) and Ma et al. (2015).

A biomarker is an objectively measurable characteristic that is able to detect abnormal biological processes, pathogenic processes and/or pharmacological responses to therapy (Bazazeh, Shubair, & Malik, 2016). To the best of our knowledge, there is no accepted definitive biomarker of Parkinson's disease. The development of early diagnostic biomarkers can lead towards more promt clinician's intervention at the onset of disease and can aid monitor the progress of medications therapy interventions that may slow or stop the course of the disease (Miller & O'Callaghan, 2015). Effors are made in the machine learning and data mining community to discover biomarkers distinguishing idiopathic Parkinson's disease patients. Research includes detection of possible biomarkers from image (Adeli et al., 2017; Goebel et al., 2011; L. Liu et al., 2018; Singh & Samavedham, 2015), speech (Tsanas, Little, McSharry, Spielman, & Ramig, 2012), and biochemical data (Alberio et al., 2013).

Tremor is one of the symptoms strongly associated with Parkinson's disease. Several methods for numerical assessment of the intensity of tremor have been proposed. These methods include time series analysis (Timmer et al., 1993), spectral analysis (Riviere, Reich, & Thakor, 1997) and analysis with adaptive Fourier modeling (Riviere et al., 1997) and they address tremor detection and quantification. Recent works are based on body fixed sensors (BFS) for long-term monitoring of patients (Patel et al., 2009).

In the course of their disease, patients are prescribed antiparkinson medications therapies in order to control the troubling symptoms. As the disease progresses, the medications treatment can become ineffective and—as a last resort—clinicians use deep brain stimulation (DBS) surgery to control the Parkinson's disease symptoms. Data mining research confirms that DBS significantly improves the patients' motor function (Y. Liu et al., 2014). Depending on the chosen method for DBS, a great reduction in dose of medication, or conservation of cognitive functions can be achieved. In order to predict the neurological effects related to different electrode-contact stimulation, Szymański, Kubis, and Przybyszewski (2015) tracked the connections between the stimulated part of the subthalamic nucleus and the cortex with the help of diffusion tensor imaging (DTI).

Tsanas (2012) addresses the progression of Parkinson's disease for 42 patients over a six month period by predicting the total score of patients' motor symptoms severity and the total UPDRS (Unified Parkinson's Disease Rating Scale) score using linear and nonlinear regression techniques. Eskidere et al. (2012) build on previous work (Tsanas, Little, McSharry, & Ramig, 2010; Tsanas, Little, et al., 2010) by proposing and comparing

the performance of multiple regression methods for predicting the above mentioned total scores. It is worth noting that during their involvement in the clinical study for data collection, patients did not receive their antiparkinson medications.

While clustering usually focuses on patient grouping with the aim of diagnosing new patients, none of the listed methods follows the progression of the disease, and to the best of our knowledge, no data mining research in the field of Parkinson's disease analyzed the development of the disease in combination with the medications that the patients receive. Identification of groups of patients based on the similarity of their symptoms and the clinicians' reaction with medications modification in order to keep the patients as stable and in good status as possible, can be helpful in the assignment of personalized therapies and adequate patient treatments. For that purpose, we propose a methodology for identification of groups of patients based on the severity of their symptoms, determination of disease progression, and the consequent patterns of medications modifications.

In the context of the PD_manager project (PD_manager: m-Health platform for Parkinson's disease management, 2015), Mileva-Boshkoska et al. (2017) developed a decision support model for Parkinson's disease medication changes. Models are developed in collaboration with Parkinson's disease experts and no data mining is used in the process.

Chapter 3

Parkinson's Disease Data

The data we used in this thesis is a part of the Parkinson's Progression Markers Initiative (PPMI) funded by the Michael J. Fox Foundation. The PPMI study is comprised of neurological and movement disorder study sites throughout the United States, Europe, and Australia, and study cores with expertise in clinical assessments, neurological imaging, biologic sample storage (biorepository), bioinformatics, statistics, bioanalytics, and genetics (Marek et al., 2011). The PPMI data set collection includes clinical assessments and evaluations, subject demographics, imaging data, and biological samples. In addition to these data, the PPMI study also keeps track of all the concomitant medications the Parkinson's disease patients are receiving throughout their involvement in the study. This chapter presents a short description of symptoms and medications data used in our analyses to determine indicators of changes in the patients' quality of life—important symptoms and reactions to these symptoms by medications therapy modifications. Section 3.1 presents a short description of symptoms data used by clinicians to follow the quality of life of their patients. In Section 3.2 we present groups of antiparkinson medications used in the therapy of Parkinson's disease patients. We describe how these medications are recorded in the PPMI study and how we relate them to symptoms reflecting the overall status of the patients.

3.1 Symptoms Data

Parkinsonism is a term that covers a range of conditions that have similar symptoms to Parkinson's. About 85% of people with parkinsonism have Parkinson's (sometimes called idiopathic Parkinson's) (European Parkinson's Disease Association, 2016). The quality of life of patients suffering from Parkinson's disease is monitored by a set of clinical symptoms. These symptoms describe different aspects of patients' everyday life and are obtained and evaluated by standardized questionnaires. The most popular and widely accepted questionnaire for determining the medical condition and the quality of life of a patient suffering from Parkinson's disease is the Movement Disorder Society (MDS) sponsored revision of Unified Parkinson's Disease Rating Scale (MDS-UPDRS) (Goetz et al., 2008). It is a questionnaire consisting of 65 questions concerning the progression of disease symptoms. MDS-UPDRS is divided into four parts. Part I consists of questions about the 'non-motor experiences of daily living'. These questions address complex behaviors, such as hallucinations, depression, apathy, etc., and patient's experiences of daily living, such as sleeping problems, daytime sleepiness, urinary problems, etc. Part II expresses 'motor experiences of daily living'. This part of the questionnaire examines whether the patient experiences speech problems, the need for assistance with the daily routines such as eating or dressing, etc. Part III is referred to as the 'motor examination' and collects data about the

Table 3.1: Short overview of the patients whose data are used in the experimental work.

Characteristic	Value
Number of patients	405
Male/Female patients	265/140
Age range on baseline visit	33–84
Average age	61
Visit range	1–5
Average number of visits	3.32
Number of instances	1335

motor symptoms that are typical for Parkinson's disease patients and involve bradykinesia (slowness of movement), rigidity, tremor, postural instability, etc. Part IV concerns 'motor complications', which are mostly developed when the main antiparkinson drug levodopa is used for a longer time period.

Each question from the MDS-UPDRS questionnaire is anchored with five responses that are linked to commonly accepted clinical terms: 0 = normal (patient's condition is normal, symptom is not present), 1 = slight (symptom is present and has a slight influence on the patient's quality of life), 2 = mild, 3 = moderate, and 4 = severe (symptom is present and severely affects the normal and independent functioning of the patient, i.e. her/his quality of life is significantly decreased). In terms of data science, questions from the questionnaire can be considered as attributes in a data table, while responses to the questions provide values of the corresponding attributes. The evaluation of MDS-UPDRS symptoms is performed periodically, approximately every 3–6 months, throughout the patients' 5-year involvement in the study, thus providing a longitudinal view and data about the progression of Parkinson's disease of each of the involved patients. We refer to the time points when symptoms are updated as *visits* as these are actual visits of the patients.

Note that PPMI data collection holds data about patients with varying times from diagnosis and length of involvement in the PPMI study. There are patients who have only recently been diagnosed with Parkinson's disease as well as patients with almost 2 years passed between their diagnosis and the start of their involvement in the PPMI study. Also, there are patients who have only started their involvement in the PPMI study and those who have concluded theirs. The experimental data include symptoms and medications data of 405 Parkinson's disease patients from the PPMI study. Table 3.1 presents a short overview of the patients whose data are used in our experimental work. The data cover the status of 265 male and 140 female patients. At the beginning of their involvement in the study (baseline visit), the youngest patient was 33 years old and the oldest patient was 84 years old. The average age of patients is 61 years. Each patient has made 1 to 5 visits to the clinician. The average number of recorded visits is 3.32. The experimental data consist of 1,345 patients' visits and each visit is considered a separate data instance, representing the basic building block of the methodology described in Chapters 5 and 6.

The cognitive state of Parkinson's disease patients is assessed by the Montreal Cognitive Assessment (MoCA) (Dalrymple-Alford et al., 2010) questionnaire. It is a rapid screening instrument for mild cognitive dysfunction. It is a 30 point questionnaire consisting of 11 questions, designed to assess different cognitive domains: attention and concentration, executive functions, memory, language, visuoconstructional skills, conceptual thinking, calculations, and orientation. The cognitive evaluation of patients using the MoCA questionnaire is done on every other visit, starting from the fourth visit. In addition to MoCA,

Table 3.2: Characteristics of the questionnaire data used in the analysis.

Questionnaire	Number of questions	Answers value range	Higher value indicates higher symptom severity
MDS-UPDRS Part I	6	0–4	Yes
MDS-UPDRS Part Ip	7	0–4	Yes
MDS-UPDRS Part II	13	0–4	Yes
MDS-UPDRS Part III	35	0–4	Yes
MoCA	11	0–1	No
PASE	7	1–2	No
SCOPA-AUT	21	0–3	Yes
QUIP	11	0–1	Yes

physicians also use the Questionnaire for Impulsive-Compulsive Disorders in Parkinson's disease (QUIP) (Weintraub et al., 2012) to address four major and three minor impulsive-compulsive disorders. The patient is impulsive if he/she has impulsion problems with any of the considered impulsive compulsive problems. In our research, we use QUIP to determine whether the patient has an impulsivity problem or not. Questions are not considered as separate attributes in the data analysis. QUIP is administered regularly on every other visit to the clinician.

Scales for Outcomes in Parkinson's disease–Autonomic (SCOPA-AUT) is a specific scale to assess autonomic dysfunction in Parkinson's disease patients (Visser, Marinus, Stiggelbout, & Van Hilten, 2004). It consists of 23 questions concerning gastrointestinal, urinary, cardiovascular, thermoregulatory, pupillomotor, and sexual symptoms. The sexual symptoms are gender-specific and were therefore omitted from our analysis, thus bringing the number of considered SCOPA-AUT attributes to 21. Each question from the SCOPA-AUT questionnaire is anchored with four responses that are linked to commonly accepted clinical terms: 0 = never (patient has never experienced the particular symptom), 1 = sometimes (symptom has occurred sometimes), 2 = regularly, 3 = often (patient often has problems with the particular symptom). Urinary symptoms can also be evaluated with value 9, indicating that the patient is wearing a catheter. Responses to the SCOPA-AUT questionnaire are updated on every other visit.

Physical Activity Scale for the Elderly (PASE) (Washburn, Smith, Jette, & Janney, 1993) is a questionnaire that is a practical and widely used approach for physical activity assessment in epidemiologic investigations. The PASE score combines information on leisure, household, and occupational activity. Washburn et al. (1993) states that the PASE test-retest reliability coefficient (0.75) exceeded those reported for other physical activity surveys. Responses to the PASE questionnaire are updated on every other visit, starting from the fourth visit. The periodical update of patients symptoms in the PPMI study allows the clinicians to monitor patients' disease development through time. As mentioned above, answers to the questions from each questionnaire form the vectors of attribute values.

Table 3.2 presents a summary of the symptoms data sets considered in our research. It lists the number of considered questions from each questionnaire, the range of attribute values, and the nature of the attribute values. All considered questions have ordered values and with the exception of questions from MoCA and PASE, larger values suggest higher symptom severity and decreased quality of life of PD patients.

Symptoms of patients suffering from Parkinson's disease are grouped into several data sets, representing distinct views of the data. These views consist of data from MoCA test, motor experiences of daily living, non-motor experiences of daily living, complex motor

examination data, etc. For each patient these data are obtained and updated periodically (on each or every second patient's visit to the clinician), usually first at the beginning of the patient's involvement in the PPMI study, and then approximately every 6 months, in total duration of 5 years—providing the clinicians with the opportunity to follow the development of the disease. Visits of each patient can be viewed as time points, and the collected data on each visit is the data about the patient at the respective time point. All time points collected for one patient form a short time series.

When considering the possibility of using a multi-view framework, the independence of separate views should be inspected. The conditional independence of separate views imposes constraints on their shared latent representation. If the conditional independence constraint is respected, it can improve the quality of a learned low dimensional representation thus leading towards improved learning results (White, Zhang, Schuurmans, & Yu, 2012). In their work, Goetz et al. (2008) state that MDS-UPDRS shows high internal consistency (Cronbach's alpha = 0.79–0.93 across the MDS-UPDRS parts (described above), indicating acceptable to excellent internal consistency). Cronbach's alpha (Cronbach, 1951) is a measure of internal consistency, measuring how closely related a set of items are as a group. MDS-UPDRS across-part correlations range from 0.22 to 0.66, indicating not acceptable or poor internal consistency. Across-part correlations determine the correlation between pairs of MDS-UPDRS parts. Results confirm that each part assesses a different aspect of PD. Reliable factor structures for each part are obtained (comparative fit index > 0.90 for each part), which supports the use of sum scores for each part, when compared to using a total score of all parts. Factor analysis is a strong clinical/statistical method for scale evaluation (Goetz et al., 2008). It examines whether items can be clustered and allows clinicians to determine if these clusters fall into components that represent clinically relevant domains (Goetz et al., 2008).

3.2 Medications Data

The PPMI data collection offers information about all concomitant medications that patients used during their involvement in the study. These medications are described with names, medical conditions they are prescribed for, and time when the patient started and (if) ended the medications therapy. For the purpose of our research, we initially concentrate only on whether the patient receives a therapy with antiparkinson medications, and which combination of antiparkinson medications the patient has received between the time points when the MDS-UPDRS test and the MoCA test were administered. The main families of drugs used for treating motor symptoms are levodopa, dopamine agonists, and MAO-B inhibitors (National Collaborating Centre for Chronic Conditions, 2006). Medications that treat Parkinson's disease-related symptoms but are not from the above-mentioned groups of medications are referred to as *other*.

The most widely used treatment for Parkinson's disease for over 30 years is levodopa. It crosses the blood/brain barrier and once it is in the nervous system, it is transformed into dopamine in the dopaminergic neurons by dopa-decarboxylase. Motor symptoms are produced by a lack of dopamine in the substantia nigra, so levodopa is used to temporarily diminish the motor symptomology. The chronic administration of levodopa in the treatment of Parkinson's disease can cause several serious side effects, that include on/off fluctuations and levodopa-induced dyskinesia. When this occurs, Parkinson's disease patients change rapidly from a state with good response to medication and few symptoms ("on" state) to a status with no response to medication and important motor symptoms ("off" state). In order to omit or prolong the time before these side-effects occur, levodopa doses are kept as low as possible while maintaining functionality of patients. A common

practice is to delay the initiation of levodopa therapy and use alternatives for some time.

Dopamine agonists were initially used for patients experiencing on-off fluctuations and dyskinesias as a complementary therapy to levodopa. Now they are mainly used on their own as an initial therapy for motor symptoms with the aim of delaying motor complications. Dopamine agonists are known to cause side effects which include insomnia, hallucinations, constipation, and problems with impulsive control. The existence of these side effects forces clinicians to change patients' medication therapies either by decreasing the dosage or stopping the dopamine agonist therapy.

MAO-B inhibitors increase the level of dopamine in the basal ganglia by blocking its metabolization and provide increased levels of levodopa in the striatum (nuclei in the forebrain and a critical component of the motor and reward system). Similar to dopamine agonists, MAO-B inhibitors are used in the first stages of the disease to control the motor symptoms and delay the need for taking levodopa. Therapies with MAO-B inhibitors are also used for the treatment of depression and anxiety. Both symptoms can be experienced by Parkinson's disease patients.

Clinicians follow the patients' status and respond to changes in modifications of medications. The modifications can be in terms of increasing or decreasing the dosage of a certain group of antiparkinson medications, introducing or removing medications from patients' therapies, changing the frequency and dosage of medication intake, etc. Clinicians can modify the patients' medications at any time and their decision is not strictly connected to the visits at which the above-mentioned questionnaires are administered. Many times clinicians do phone call-ups to patients in order to stay informed of the patients' status and how the disease progresses, and if necessary, they modify the therapy in order to control and stabilize the status of the patients. The medications therapy for Parkinson's disease patients is highly personalized. Patients take different medications with personalized plans of intake. In order to be able to compare different therapies, dosages of Parkinson's disease medications are translated into a common score called Levodopa Equivalent Daily Dosage (LEDD).

Figure 3.1 presents scenarios of medications dosage changes of a single patient. Points V1, V2, V3, and V4 present time points on the continuous timeline with visits 1, 2, 3, and 4 to the clinician. The red lines represent the intake duration of each antiparkinson medication, D1, D2, and D3. The line width indicates the dosage of the medications, where wider line width indicates increased dosage. As evident from Figure 3.1, on visit 1, the patient receives a therapy consisting of medications D1 and D3. Sometime between visits 1 and 2 the therapy is updated by introducing medication D2 and removing medication D1. The snapshot of medications therapy in visit 3 shows that the patient's therapy is modified by increasing the LEDD of medication D2 and consists of medications D2 and D3. By visit 4, the medications therapy consists only of medication D3. As mentioned above, the clinicians follow the status of the patient and based on the overall status decide how to modify the therapy. For example, if certain motor symptoms are getting overly problematic, clinicians will try to introduce levodopa based medications or increase their dosage. However, if the patient is experiencing hallucinations or has problems with impulse control and takes dopamine agonists, clinicians will decrease the dosage of these medications or remove them from the therapy.

We preprocessed the medications data presented in the PPMI concomitant medications log by recording only antiparkinsonian medications and connecting the medications and their dosages to the patients' visits. For example, for the patient from Figure 3.1, we would record that in visit 1 the patient was taking medications D1 and D3 and as well as their respective LEDD values. For visit 2 we would record that the therapy consisted of medications D2 and D3, etc.

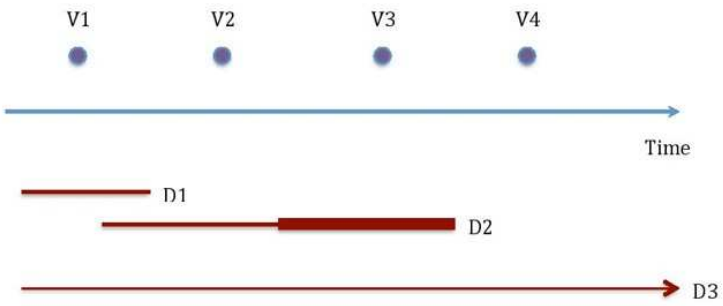


Figure 3.1: Example of Parkinson's disease patient therapy modifications between visits 1 and 4. The blue line presents the linear timeline, while points V1, V2, V3, and V4 present four consecutive visits to the clinician when the MDS-UPDRS questionnaire is administered. The red lines present the duration of intake for each antiparkinson medication, while the line width presents the dosage of the medication.

Chapter 4

Descriptive Rule Learning

Rule learning algorithms typically proceed in two phases: rule refinement selects terms for specializing the rule, and rule selection selects the final rule among several candidates. While most conventional algorithms use the same heuristic in both phases, recent research (Stecher et al., 2014) indicates that two separate heuristics improve the coverage of positive examples, and may result in better classification accuracy. This chapter presents and evaluates two new beam search rule learning algorithms: DoubleBeam-SD for subgroup discovery and DoubleBeam-RL for classification rule learning. The algorithms use two separate beams and can use different heuristics in the rule refinement and rule selection phase. The chapter is divided into two sections. We first introduce the problem description and then present the published *Expert Systems with Applications* journal paper that addresses the described problem.

4.1 Problem Description

Most data mining techniques aim at optimizing the predictive performance of the induced models. However, in order for these models to be utilized by an expert system or to offer decision support, their comprehensibility is of the ultimate importance. For example, in medical applications, clinicians are interested in the symptoms, conditions, and circumstances causing a certain recommendation. Other examples of application areas in need of transparent models include law, finance and knowledge discovery (Bibal & Frénay, 2016).

Rule learning is a symbolic data analysis technique that can construct comprehensible models or patterns describing the data (Clark & Niblett, 1989; Fürnkranz et al., 2012; Michalski, 1969). Compared to statistical learning techniques, the key advantage of rule learning is its simplicity and humanly comprehensible outputs, therefore, the development of new rule learning algorithms for constructing understandable models and patterns is of great interest of the data mining community.

Classification rule learning is a technique for predictive induction. Models consist of IF-THEN rules covering the entire problem space. The rule generation process is performed on the labeled data with the intention to construct rules with high predictive power, covering as many as possible positive examples and as few as possible negative examples. Contrary to classification rule learning, subgroup discovery (Atzmüller, 2015; Klösgen, 1996; Wrobel, 1997) is a descriptive induction technique where the ultimate goal is to induce individual rules describing interesting subgroups which have a significantly different class distribution to that of the entire population (Klösgen, 1996; Wrobel, 1997). Models are generated on the labeled data, however, they consist only of rules describing the properties of individual groups of target class instances.

A common property of classification rule learning and subgroup discovery is the rule

construction process. In both cases, rule construction is performed in two phases: the rule refinement and the rule selection phase. Typically, different types of heuristics are used for classification rule induction and subgroup induction. Researchers usually choose one heuristic and use the same heuristic in the two phases of the rule construction process: (i) a heuristic is used to evaluate *rule refinements*, i.e. to select which of the refinements (specializations) of the current rule will be further explored, and (ii) the same heuristic is used in *rule selection* to decide which of the constructed rules will be added to the rule set.

Since one of the goals of rule learning is to optimize the predictive performance of constructed rules, heuristics used in the rule construction process are customized and adapted for the selection phase. Given the different natures of the refinement and selection phase, this practice may not fully take advantage of each phase and not lead to the best discovered rules. Stecher et al. (2014) described this divergence and proposed to use separate heuristics for each of the two rule construction phases. They suggested that in the refinement phase, so-called *inverted heuristics* should be used for evaluating the relative gain obtained by refining the current rule. The key idea of these heuristics is that while most conventional rule learning heuristics, such as the Laplace or the m-estimate, anchor their evaluation on the empty rule that covers all examples, inverted heuristics anchor the point of view on the base rule, which is more appropriate for a top-down refinement process. In terms of the formulas for the conventional rule learning heuristics, it means that the values of P and N are constant and denote the total number of positive and negative examples in the data set. In the *inverted heuristics* setup, these values change, and represent the number of positive and negative examples covered by the parent rule of the rule that is being evaluated.

The introduction of separate heuristics that take full advantage of the different natures of the rule construction phases can lead towards discovering better rules in terms of their classification power and their coverage. Stecher et al. (2014) mention that longer rules are an expected side effect of using *inverted heuristics* in the refinement phase which might be welcome in the analysis of medical data, as longer rules have a greater descriptive power and are preferred in some application areas such as medicine.

We explored the possibility of separating the phases of the rule learning process in two settings: classification rule learning and subgroup discovery. We propose two new algorithms, the DoubleBeam-RL algorithm for classification rule learning and the DoubleBeam-SD algorithm for subgroup discovery. The separation of the refinement and selection phase in the rule learning process was done two-fold by introducing separate heuristics and using separate beams for rule refinement and selection.

The introduction of two beams in the rule learning process effectively widens the search space and allows the algorithms to construct and detect rules which could be overlooked otherwise. This separation keeps track of the set of rules which may have low selective quality—they cover some negative examples, but have high refinement quality—rule also has a high coverage of positive examples. It also keeps track of rules that have very high selective quality but only cover a fraction of positive examples. By keeping track of the best rules for both refinement and selection we extend the possibility that by the end of the search process the algorithm will find rules of better quality.

We compare the performance of the DoubleBeam-RL algorithm and the DoubleBeam-SD algorithm to their state-of-the-art counterparts. In order to omit any bias introduced by the algorithms' parameters, we introduce a double-loop approach for parameter setting. In the classification rule learning setting, the experimental results confirm previously shown benefits of using two separate heuristics for rule refinement and rule selection. In subgroup discovery, DoubleBeam-SD algorithm variants outperform several state-of-the-art related algorithms.

4.2 Related Publication

The rest of this chapter presents the *Expert Systems with Applications* journal paper.

Publication related to this contribution

Journal Paper

Valmarska, A., Lavrač, N., Fürnkranz, J., & Robnik-Šikonja, M. (2017). Refinement and selection heuristics in subgroup discovery and classification rule learning. *Expert Systems with Applications*, *81*, 147–162. doi:10.1016/j.eswa.2017.03.041

This publication contains the following contributions:

- We present an overview of existing algorithms for classification rule learning and subgroup discovery.
- We present the process of rule learning in the coverage space and motivate the separation of the refinement and selection phase of the learning phases.
- We present the DoubleBeam-RL algorithm for classification rule learning and empirically compare its performance to the performance of the CN2 algorithm (Clark & Niblett, 1989), the Ripper algorithm (Cohen, 1995), and the best performing algorithm from (Stecher et al., 2014).
- We show that our algorithm performs comparably to its state-of-the-art counterparts.
- We confirm the advantage of separating the learning phases in both the classification rule learning setting as well as in the subgroup discovery setting.
- We present the DoubleBeam-SD algorithm for subgroup discovery and present three approaches for instance weighting.
- We compare its performance to the SD algorithm (Gamberger & Lavrač, 2002), the CN2-SD algorithm (Lavrač et al., 2004), and the APRIORI-SD algorithm (Kavšek et al., 2003). We empirically show that the DoubleBeam-SD algorithm performs comparably to the considered state-of-the-art algorithms for subgroup discovery. We also show that a variant of this algorithm produces rules that are statistically more unusual than the rules generated by the SD and the APRIORI-SD algorithm.

The authors' contributions are as follows. The algorithms were designed and developed by Anita Valmarska with the insights from Marko Robnik-Šikonja. Anita Valmarska implemented the algorithms and performed the experimental work. Marko Robnik-Šikonja initiated the idea for double-loop cross-validation determination of default parameters and supervised the implementation of the algorithms. Nada Lavrač directed our attention towards the work of Stecher et al. (2014) and motivated the usage of inverted heuristics in the rule learning process. Johannes Fürnkranz provided helpful insights into the performance of the algorithms. All authors contributed to the text of the manuscript.

Expert Systems With Applications 81 (2017) 147–162

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Refinement and selection heuristics in subgroup discovery and classification rule learning

Anita Valmarska[a,b,*], Nada Lavrač[a,b,c], Johannes Fürnkranz[d], Marko Robnik-Šikonja[e]

[a] Jožef Stefan Institute, Jamova 39, Ljubljana, Slovenia
[b] Jožef Stefan International Postgraduate School, Jamova 39, Ljubljana, Slovenia
[c] University of Nova Gorica, Vipavska 13, Nova Gorica, Slovenia
[d] TU Darmstadt, Darmstadt, Germany
[e] University of Ljubljana, Faculty of Computer and Information Science, Slovenia

ARTICLE INFO

Article history:
Received 7 June 2016
Revised 17 March 2017
Accepted 18 March 2017
Available online 21 March 2017

Keywords:
Rule learning
Subgroup discovery
Inverted heuristics

ABSTRACT

Classification rules and rules describing interesting subgroups are important components of descriptive machine learning. Rule learning algorithms typically proceed in two phases: rule refinement selects conditions for specializing the rule, and rule selection selects the final rule among several rule candidates. While most conventional algorithms use the same heuristic for guiding both phases, recent research indicates that the use of two separate heuristics is conceptually better justified, improves the coverage of positive examples, and may result in better classification accuracy. The paper presents and evaluates two new beam search rule learning algorithms: DoubleBeam-SD for subgroup discovery and DoubleBeam-RL for classification rule learning. The algorithms use two separate beams and can combine various heuristics for rule refinement and rule selection, which widens the search space and allows for finding rules with improved quality. In the classification rule learning setting, the experimental results confirm previously shown benefits of using two separate heuristics for rule refinement and rule selection. In subgroup discovery, DoubleBeam-SD algorithm variants outperform several state-of-the-art related algorithms.

1. Introduction

While most data mining techniques aim at optimizing predictive performance of the induced models, their comprehensibility is of ultimate importance for expert systems and decision support. Examples of application areas in need of transparent models include medicine, law, finance and knowledge discovery (Bibal & Frénay, 2016).

Rule learning is a symbolic data analysis technique that can be used to construct understandable models or patterns describing the data (Clark & Niblett, 1989; Fürnkranz, Gamberger, & Lavrač, 2012; Michalski, 1969). As one of the standard machine learning techniques it has been used in numerous applications. Compared to statistical learning techniques, the key advantage of rule learning is its simplicity and human understandable outputs. Therefore, the development of new rule learning algorithms for constructing understandable models and patterns is in the core interest of the data mining community.

Symbolic data analysis techniques can be divided into two categories. Techniques for *predictive induction* produce models, typically induced from labeled data, which are used to predict the label of previously unseen examples. The second category consists of techniques for *descriptive induction*, where the aim is to find comprehensible patterns, typically induced from unlabeled data. There are also descriptive induction techniques that learn descriptive rules from labeled data, which are referred to as *supervised descriptive rule discovery* techniques (Kralj Novak, Lavrač, & Webb, 2009). Typical representatives of these techniques are subgroup discovery (SD) (Atzmueller, 2015; Klösgen, 1996; Wrobel, 1997), contrast set mining (CSM) (Bay & Pazzani, 2001), and emerging pattern mining (EPM) (Dong & Li, 1999) techniques. For instance, the task of subgroup discovery is to find interesting subgroups in the population, i.e. subgroups that have a significantly different class distribution than the entire population (Klösgen, 1996; Wrobel, 1997). The result of subgroup discovery is a set of individual rules, where the rule consequence is a class label.

An important characteristic of subgroup discovery is that its task is a combination of predictive and descriptive rule induction.

* Corresponding author.
E-mail addresses: anita.valmarska@ijs.si (A. Valmarska), nada.lavrac@ijs.si (N. Lavrač), fuernkranz@informatik.tu-darmstadt.de (J. Fürnkranz), marko.robnik@fri.uni-lj.si (M. Robnik-Šikonja).

http://dx.doi.org/10.1016/j.eswa.2017.03.041
0957-4174/

It provides understandable descriptions of subgroups of individuals which share a common target property of interest. This feature of subgroup discovery has inspired many researchers to investigate new methods that will be more effective in finding interesting patterns in the data. Most subgroup discovery approaches build on classification algorithms, e.g., EXPLORA (Klösgen, 1996), MIDOS (Wrobel, 1997), SD (Gamberger & Lavrač, 2002), CN2-SD (Lavrač, Kavšek, Flach, & Todorovski, 2004), and RSD (Lavrač, Železný, & Flach, 2002), or on algorithms for association rule learning, e.g., APRIORI-SD (Kavšek, Lavrač, & Jovanoski, 2003), SD-MAP (Atzmüller & Puppe, 2006), and Merge-SD (Grosskreutz & Rüping, 2009).

The main difference between classification rule learning and subgroup discovery is that subgroup discovery algorithms construct individual rules describing the properties of individual groups of target class instances, while classification rule learning algorithms construct a set of classification rules covering the entire problem space.

A common property of classification rule learning and subgroup discovery is that rule construction is performed in two phases: the rule refinement and the rule selection phase. Typically, different types of heuristics are used for classification rule induction and subgroup induction. Researchers usually choose one heuristic and use the same heuristic in the two phases of the rule construction process: (i) a heuristic is used to evaluate *rule refinements*, i.e. to select which of the refinements (specializations) of the current rule will be further explored, and (ii) the same heuristic is used in *rule selection* to decide which of the constructed rules will be added to the rule set. For learning classification rules, Stecher, Janssen, and Fürnkranz (2014) proposed to use separate heuristics for each of the two rule construction phases, and suggested that in the refinement phase, so-called *inverted heuristics* should be used for evaluating the relative gain obtained by refining the current rule. The key idea of these heuristics is that while most conventional rule learning heuristics, such as the Laplace or the *m*-estimate, anchor their evaluation on the empty rule that does not cover any examples, inverted heuristics anchor the point of view on the base rule, which is more appropriate for a top-down refinement process.

In this paper, we test the utility of inverted heuristics in the context of subgroup discovery as well as in the context of classification rule learning. For this purpose we have developed two new beam search rule learning algorithms, named DoubleBeam-SD for subgroup discovery and DoubleBeam-RL for classification rule learning, respectively. The algorithms allow to combine various heuristics for rule refinement and rule selection, with the goal of determining their optimal combination, and, in consequence, learn rules with better coverage and better descriptive power without compromising rule accuracy. The introduction of two separate beams enlarges the search space, enabling the learner to find rule sets that are more accurate as well as more interesting to the end user. For example, physicians appreciate rules that are highly accurate when used in patient classification, but prefer understandable rules that precisely characterize the patients in terms of the features that distinguish the patients from the control group.

We compare the double beam search algorithms to state-of-the-art subgroup discovery and rule learning algorithms by experimentally evaluating them on the UCI data sets, using the same data sets as in previous research of Stecher et al. (2014). All the competitors are used with their default parameters from their corresponding software platforms. In order to determine useful default configurations for our algorithms, we employ a data set hold-out methodology for parameter setting with the goal of finding the optimal configuration without tuning the algorithms to a particular data set.

The rest of this paper is organized as follows. Section 2 provides the necessary background on rule learning and subgroup discovery, followed by the introduction of the coverage space and an illustrative example, explaining the advantages of using inverted heuristics in rule refinement. It also summarizes the findings of Stecher et al. (2014) concerning the use of inverted heuristics in rule learning. Section 3 is concerned with subgroup discovery presenting the DoubleBeam-SD algorithm and its variants, followed by a description of the experimental setting and the obtained results. Section 4 outlines the DoubleBeam-RL algorithm for classification rule learning, followed by a description of the experimental setting, and the presentation of experimental results. Finally, Section 5 presents the conclusions and ideas for further work.

2. Rule learning: background and related work

Rule learning is a standard symbolic data analysis technique used for constructing understandable models and patterns. Its main advantage over the other data analysis techniques is its simplicity and comprehensibility of its outputs. Rule learning has been extensively used both in predictive and descriptive rule learning settings, where by applying different rule evaluation heuristics different trade-offs between the consistency and coverage of constructed rules can be achieved.

This section first presents a short overview of classification rule learning and subgroup discovery. It introduces the coverage space used as a tool for studying the properties of different heuristics and presents the idea of using two separate heuristics for rule refinement and rule selection illustrated on a selected UCI data set. The section ends with the description of closely related work regarding the use of inverted heuristics in classification rule learning.

2.1. Classification rule learning

The task of classification rule learning is to find models which would ideally be *complete* (cover all positive examples, or at least most of the positives), and *consistent* (not cover any negative examples, or at most a very small number of negatives). Multi-class classification problems can be dealt with by using the one-versus-all approach, which learns one rule set for each class, where the examples labeled with the chosen class are considered as positive target class examples, and all examples of other classes as negatives.

There are numerous classification rule learning algorithms, the most popular being AQ, CN2 and Ripper. The AQ algorithm (Michalski, 1969), which was the first to propose the covering algorithm for rule set construction, is a top-down beam search algorithm that uses a random positive example as a seed for finding the best rule. The CN2 algorithm (Clark & Niblett, 1989) combines the ideas from the AQ algorithm and the decision tree learning algorithm ID3 (Quinlan, 1983), given the similarity of rule learning to learning decision trees, where each path from the root of the tree to a tree leaf can be viewed as a separate rule. It constructs an ordered decision list by learning rules describing the majority class examples in the training set. Once the learned rule is added to the decision list, all the covered examples, both positive and negative, are removed from the training data set, and the rule induction process is continued on the updated training set. Ripper (Cohen, 1995) is the first rule learning algorithm that effectively overcomes the overfitting problem and is thus a very powerful rule learning system. The algorithm constructs rule sets for each of the class values. Initially, the training data set is divided into a *growing* and a *pruning* set. Rules are learned on the growing set, and then pruned on the *pruning* set by incrementally reducing the error rate on the pruning set. A pruned rule is added to the rule set if the description length of the newly constructed rule set is at most *d* bits longer (a parameter) than the already induced rule set. Otherwise, the rule learning process is stopped. Similarly to the CN2

Table 1
Comparison of the DoubleBeam-RL algorithm to the state-of-the-art classification rule learners CN2, Ripper and SC-ILL.

Algorithm	Type of search	Separate refinement heuristic	Stopping criterion	Rule pruning	Post-processing
CN2	Beam	No	No beam improvement	No	No
Ripper	Greedy	No	MDL	Yes	Yes
SC-ILL	Greedy	Yes	No negative examples covered	No	No
DoubleBeam-RL	Beam	Yes	*maxSteps*	No	No

algorithm, when a new rule is added to the rule set, all the instances covered by that rule are removed from the growing set. In addition to pruning the rules before adding them to induced rule set, Ripper prevents rules overfitting in a post-processing phase in which the learned rule set is optimized and the selected rules are re-learned in the context of the other rules. FURIA (Hühn & Hüllermeier, 2009) is a classification rule learning algorithm which extends the Ripper algorithm by learning fuzzy rules.

Despite its long history, rule learning is still actively researched and routinely applied in practice. For example, Napierala and Stefanowski (2015) use rule learning with argumentation to tackle imbalanced data sets, and Ruz (2016) explores the order of instances in seeding rules to improve the classification accuracy. Minnaert, Martens, De Backer, and Baesens (2015) discuss the importance of proper rule evaluation measures for improving the accuracy of classification rule learning algorithms. They also introduce multi-criteria learning and investigate a Pareto front as a trade-off between comprehensibility and accuracy of rule learners.

In a line of research started by Parpinelli, Lopes, and Freitas (2002), rule learning is turned into an optimization problem using an ant colony optimization approach. The initial rule learning algorithm, named Ant-Miner, worked for nominal attributes only, but was later improved by Pičulin and Robnik-Šikonja (2014) to efficiently handle numeric attributes. Classification rule learning has been a vivid topic of research also in inductive logic programming and relational data mining. For example, Zeng, Patel, and Page (2014) developed the QuickFOIL algorithm that improves over the original FOIL algorithm (Quinlan & Cameron-Jones, 1993).

Learning rules can be regarded as a search problem (Mitchell, 1982). Search problems are defined by the structure of the search space, a search strategy for searching through the search space, and a quality function (a *heuristic*) that evaluates the rules in order to determine whether a candidate rule is a solution or how close it is to being a solution to be added to the rule set, i.e. the final classification model. The search space of possible solutions is determined by the model language bias (Fürnkranz et al., 2012). In propositional rule learning, the search space consists of all the rules of the form *targetClass* ← *Conditions*, where *targetClass* is one of the class labels, and *Conditions* is a conjunction of features. Features have the form of $A_i = v_{ij}$ (attribute A_i has value v_{ij}).

For learning a single rule, most learners use one of the following search strategies: *general-to-specific* (*top-down hill-climbing*) or *specific-to-general* (*bottom-up*), where the former is more commonly used. Whenever a new rule is to be learned, the learning algorithm initializes it with the *universal rule* $\mathbf{r}^\top$. This is an empty rule that covers all the examples, both positive and negative. In the rule refinement phase, conditions are successively added to this rule, which decreases the number of examples that are covered by the rule. Candidate conditions are evaluated with the goal of increasing the consistency of the rule while maintaining its completeness, i.e. a good condition excludes many negative examples and maintains good coverage on the positive examples.

Heuristic functions are used in order to evaluate and compare different rules. Different heuristics implement different trade-offs between these two objectives. While CN2 and Ripper use entropy as the heuristic evaluation measure, numerous other heuristic functions have been proposed in rule learning—for a variety of heuristics and their properties the interested reader is referred to Fürnkranz et al. (2012). The most frequently used heuristics in rule learning are:

Precision:

$$\mathrm{h}_{prec}(p, n) = \frac{p}{p+n} \tag{1}$$

Laplace:

$$\mathrm{h}_{lap}(p, n) = \frac{p+1}{p+n+2} \tag{2}$$

m-estimate:

$$\mathrm{h}_{m\text{-}est}(p, n, m) = \frac{p + m \cdot \frac{P}{P+N}}{p+n+m} \tag{3}$$

where, for a given rule, arguments p and n denote the number of positive and negative examples covered by the rule (i.e. the true and false positives, respectively), and P and N in Eq. (3) denote the total number of positive and negative examples in the data set. Given that these heuristics concern the problem of selecting the best of multiple refinements of the same base rule (the empty rule, universal rule), the values P and N can be regarded as constant, so that the above functions may be written as $h(p, n)$ depending only on the true and false positives.

Table 1 compares the DoubleBeam-RL classification rule learning algorithm (introduced in Section 4) to the state-of-the-art classification rule learners that were used in the experiments. CN2 and DoubleBeam-RL are beam search algorithms, while Ripper and SC-ILL are greedy algorithms, adding conditions to the rules which maximize their respective heuristics. The DoubleBeam-RL and SC-ILL algorithms use separate heuristics adapted for the refinement and selection phase of the rule learning process. Ripper is the only considered classification rule learning algorithm which employs rule pruning and optimization of rule sets in post-processing. The algorithms use different stopping criteria; for example, Ripper uses a heuristic based on minimum description length (MDL) principle.

2.2. Subgroup discovery

The goal of data analysis is not only building prediction models, but frequently the aim is to discover individual patterns that describe regularities in the data (Fürnkranz et al., 2012; Kralj Novak, Lavrač, Zupan, & Gamberger, 2005; Wrobel, 1997). This form of data analysis is used for data exploration and is referred to as *descriptive induction*. Subgroup discovery is a form of descriptive induction. The task of subgroup discovery is to find subgroups of examples which are sufficiently large while having a significantly larger distribution of target class instances than the original target class distribution.

Like in classification rule learning, individual subgroup descriptions are represented as rules in the form *targetClass* ← *Conditions*, where the *targetClass* is the target class representing the property of interest, and *Conditions* is a conjunction of features that are characteristic for a selected group of individuals.

Subgroup discovery is a special case of the more general task of rule learning. Classification rule learners have been adapted

Table 2
Some properties of subgroup discovery algorithms DoubleBeam-SD, APRIORI-SD, SD, and CN2-SD.

Algorithm	Type of search	Separate refinement heuristic	Stopping criterion	Post-processing
APRIORI-SD	Exhaustive	No	minSup, minConf	Yes
SD	Beam	No	No beam improvement	Yes
CN2-SD	Beam	No	No beam improvement	No
DoubleBeam-SD	Two beams	Yes	*maxSteps*	Optional

to perform subgroup discovery with heuristic search techniques drawn from classification rule learning. These algorithm also apply constraints, which are appropriate for descriptive rule learning. The research in the field of subgroup discovery has developed in different directions. Exhaustive methods, which include EXPLORA (Klösgen, 1996), SD-MAP (Atzmüller & Puppe, 2006) and APRIORI-SD (Kavšek et al., 2003), guarantee the optimal solution given the optimization criterion. The APRIORI-SD algorithm draws its inspiration from the association rule learning algorithm APRIORI (Agrawal & Srikant, 1994), but restricts it to constructing rules that have only the target variable (the property of interest) in their head, with *weighted relative accuracy* (WRACC), defined in Eq. (5), used as a measure of rule quality. In order to improve the inferential power of the subgroup describing rules, the APRIORI-SD algorithm uses a post-processing step to reduce the generated rules to a relatively small number of diverse rules. This reduction is performed using the weighted covering method proposed by Gamberger and Lavrač (2000). When a rule is added to the induced rule set, weights of examples covered by the rule are decreased. This allows the method to prioritize rules which cover yet uncovered examples, thus promoting the coverage of diverse groups of examples.

While the APRIORI-SD algorithm adapts the process of association rule learning to the context of subgroup discovery, the SD subgroup discovery algorithm (Gamberger & Lavrač, 2002) performs heuristic beam search, where rule quality is estimated using the generalization quotient heuristic

$$h_g(p, n, g) = \frac{p}{n+g}, \tag{4}$$

where p is the number of *true positives*, n is the *number of false positives*, and g is the *generalization parameter*. High-quality rules will cover many target class examples and a low number of non-target examples. The number of tolerated non-target examples covered by a rule is regulated by the generalization parameter. For small g, more specific rules are generated while for bigger values of g the algorithm constructs more general rules. The interpretation of the rules produced by the SD algorithm is improved using the above mentioned weighted covering method in post-processing (Gamberger & Lavrač, 2000).

CN2-SD (Lavrač et al., 2004) is a beam search algorithm, which adapts the CN2 (Clark & Niblett, 1989) classification rule learner to subgroup discovery. CN2-SD has introduced a weighted covering algorithm, where examples that have already been covered by one of the learned rules are not removed from the training data set, but instead their weights are decreased. The authors propose and compare different measures for rule evaluation. They argue that the most important measure for subgroup evaluation is *weighted relative accuracy* (WRACC), referred to as *unusualness*, defined as follows

$$\mathrm{WRACC}(p, n) = \frac{p+n}{P+N} \cdot \left(\frac{p}{p+n} - \frac{P}{P+N} \right) \tag{5}$$

This measure reflects both the rule significance and rule coverage, as subgroup discovery is interested in rules with significantly different class distribution than the prior class distribution that cover many instances. WRACC is the measure of choice in our experimental work on subgroup discovery for comparing the quality of the induced subgroup describing rules.

Subgroup discovery was used also in the context of semantic data mining. Adhikari, Vavpetič, Kralj, Lavrač, and Hollmén (2014) have explained mixture models by applying semantic subgroup discovery system Hedwig (Vavpetič, Novak, Grčar, Mozetič, & Lavrač, 2013) to structure the search space and to formulate generalized hypotheses by using concepts from the given domain ontologies.

Table 2 compares the DoubleBeam-SD algorithm (introduced in Section 3) to the state-of-the-art subgroup discovery algorithms APRIORI-SD, CN2-SD, and SD, which were used in the experiments. The latter algorithms use only a single heuristic for rule evaluation, designed to optimize the selection of best rules. The DoubleBeam-SD algorithms can use pairs of different heuristics (see Section 2.4) which can be applied to estimate rule quality in both the *refinement* and *selection* phases of the rule learning process. The DoubleBeam-SD algorithm stops the learning process after a predetermined number of steps (*maxSteps*). The SD and CN2-SD algorithms stop when there are no improvements of rules in the beam, i.e. when newly induced rules have lower quality than the rules already included in the beam. APRIORI-SD uses minimal support and coverage as the stopping criteria.

2.3. Coverage space

Fürnkranz and Flach (2005) introduced the coverage space as a formal framework for analyzing and visualizing the behavior of rule learning heuristics. The *coverage space* (Fürnkranz & Flach, 2005; Fürnkranz et al., 2012), referred to as the *PN space* when initially introduced by Gamberger and Lavrač (2002), enables us to plot the number of covered positive examples (true positives p) over the number of covered negative examples (false positives n). This results in a rectangular plot with values {0,1,... ,N} (where N is the total number of negative examples) on the horizontal axis and {0,1,... ,P} (where P is the total number of positive examples) on the vertical axis. Fig. 1 shows a coverage space visualization. The principle of coverage spaces can be used to plot individual rules, as well as entire theories or models composed of a rule set or a decision list.

There are four points of special interest in a coverage space:

- (0, 0) marks the *empty theory*, denoted by $\mathbf{r}_\perp$. This theory covers no positive and no negative example.
- (0, P) is the *perfect theory* which covers all positive and none of the negative examples.
- (N, 0) is the *opposite theory*. It covers all negative, but no positive examples.
- (N, P) is the *universal theory*, denoted by $\mathbf{r}^\top$. This theory covers all the examples, regardless of their label.

The ultimate goal of learning is to reach the point of *perfect theory* in the coverage space, i.e. the point (0, P). This will rarely be achieved in a single step. A set of rules will need to be constructed in order to achieve this objective. The purpose of heuristics used for rule evaluation is to determine how close a given rule is to this ideal point.

An isometric of a heuristic h is a line (or curve) in the coverage space that connects all points (p, n) for which $h(p, n) = c$ for some constant value c. Several properties of heuristics can be seen from

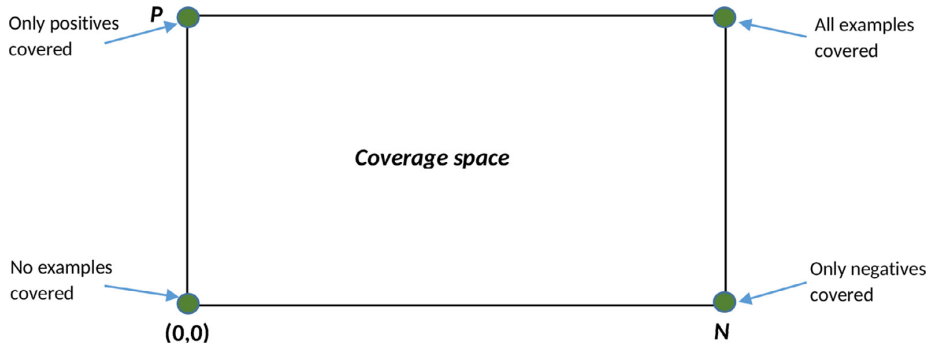


Fig. 1. Visualization of coverage space with *P* (total of positives) and *N* (total of negatives).

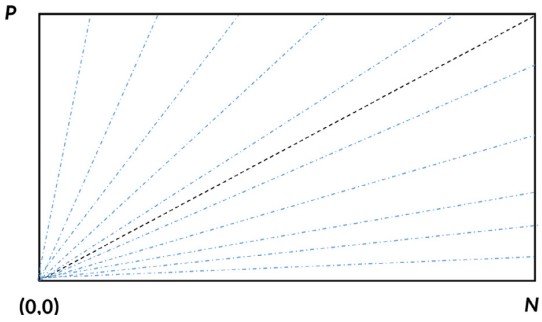


Fig. 2. Isometrics for precision.

isometrics. As an example, Fig. 2 shows the isometrics of precision, h_{prec}. These isometrics show that regarding precision all rules that cover only positive examples (points on the *P*-axis) achieve the best quality score, and all rules that cover only negative examples (points on the *N*-axis) achieve zero score. All other isometric values are obtained by rotation around the origin (0, 0) for which the value of h_{prec} is undefined. Fig. 2 presents also the disadvantage of precision, which is its inability to discriminate between rules with high and low coverage. For illustration, a rule that covers only one positive example and no negative example will have better evaluation than a rule that covers a hundred positive examples and only one negative example.

The commonly used top-down strategy for rule refinement can be viewed as a path through the coverage space. Fig. 3 illustrates rule refinement, where each point on the path corresponds to one further condition conjunctively added to the rule body. The path starts at the upper right corner, (*N, P*), with the universal rule $\mathbf{r}^{\top}$. By adding conditions to the rule, the number of covered positive and negative examples decreases and the path of the rule continues towards the origin (0, 0), which corresponds to the empty rule $\mathbf{r}_{\perp}$.

2.4. Inverted heuristics

Rule learning algorithms rely on heuristic measures to determine the quality of the induced rules. Stecher et al. (2014) propose to distinguish between rule refinement and rule selection heuristics in inductive rule learning. They argue that the nature of the separate-and-conquer rule learning algorithms opens up a possibility to use two different heuristics in the two fundamental steps of the rule learning process, i.e. rule refinement and rule selection. Using the coverage space they motivate separate evaluation of candidates for rule refinement and the selection of rules for the final theory. Stecher et al. (2014) further argue that the rule refinement step in a top-down search requires *inverted heuristics*, which can result in better rules. Such heuristics evaluate rules from the point of the current base rule, instead of the empty rule. In this way, while successively adding features to the rule (refinement), the learner favours rules with higher coverage of positive examples and thereby gives chance to rules with higher coverage to be finally selected with the selection heuristics.

Representations of the inverted heuristics in the coverage space reveal the following relationship with the basic heuristic:

$$\mathrm{q}(p,n) = \mathrm{h}(N-n, P-p) \tag{6}$$

where *p* and *n* denote the number of positive and negative examples covered by the rule, and *P* and *N* are not constant but depend on the predecessor of the currently constructed rule. For example, in the example illustrated in Fig. 5, in the first step *N* and *P* correspond to the initial top-right corner (*N, P*) in the coverage space, but when refined to rule $p \leftarrow a$, the top-right corner is moved to point ***B***. The values of *N* and *P* will change respectively. Additionally, on the refinement path, *N* and *P* will be updated with the (*N, P*) coordinates of values of point ***C***, ***D***, and ***E***, respectively in each next refinement iteration. Each of these points represent the base rule from which we observe the improvements of the consequent refinements.

Stecher et al. (2014) adapt the three standard heuristics for rule induction (introduced in Section 2.1): *precision, Laplace*, and *m-estimate*. The effect on these three heuristics is that the isometrics of their inverted variants do not rotate around the origin of the coverage space, but rotate around the point in the coverage space representing the base rule (the predecessor of the currently constructed rule). Consequently, the inverted heuristics have the following forms:

Inverted precision:

$$\mathrm{q}_{prec}(p,n) = \frac{N-n}{(P+N)-(p+n)}, \tag{7}$$

Inverted Laplace:

$$\mathrm{q}_{lap}(p,n) = \frac{N-n+1}{(P+N)-(p+n-2)}, \tag{8}$$

Inverted m-estimate:

$$\mathrm{q}_{m\text{-}est}(p,n,m) = \frac{N-n+m\cdot\frac{P}{P+N}}{(P+N)-(p+n-m)}. \tag{9}$$

The inverted heuristics are not suited for rule selection. They do favor rules with high coverage but are also tolerant to covering

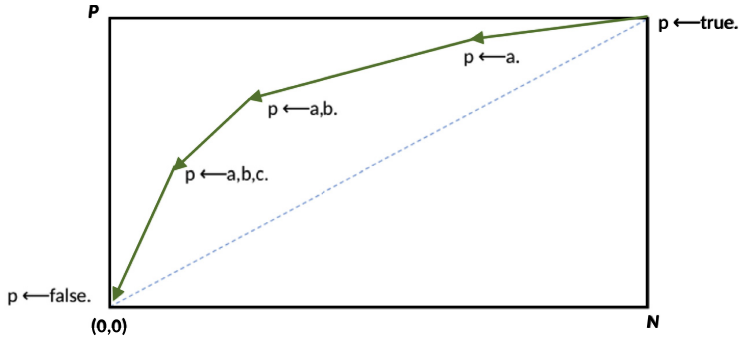


Fig. 3. A path in the coverage space of a top-down specialization of a single rule. For simplicity, a comma is used to represent the conjunction operator.

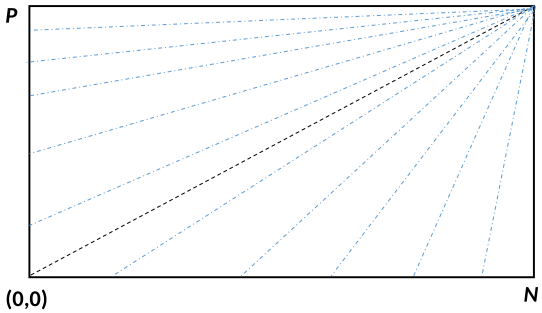


Fig. 4. Isometrics of inverted precision.

negative examples. The isometrics of inverted precision in Fig. 4 illustrate this propery.

For classification rule learning, Stecher et al. (2014) have shown that the combination of Laplace heuristic h_{lap} used in the rule selection step and the inverted Laplace heuristic $\overline{h}_{lap}$ used in the rule refinement step outperformed other combinations in terms of average classification accuracy. An interesting side conclusion from Stecher et al. (2014) is that the usage of inverted heuristics in the rule refinement phase produces on average longer rules, which are claimed to be better for explanatory purposes.

We illustrate the advantage of using inverted heuristics in the refinement phase on the UCI (Lichman, 2013) *mushroom* data set. In Fig. 5 we show the path in coverage space of top-down specialization of two rules for the class *poisonous* using different heuristics. Table 3 shows the descriptions of the coverage space points shown in Fig. 5. The red path shows the top-down specialization of a rule using the h_{lap} heuristic. From all of the refinements of the universal rule, the refinement $odor = f$ has the steepest gradient from the origin (0, 0). Therefore, this rule is selected for further refinement. However, since the number of covered positive examples is $n = 0$, the refinement process is terminated and the rule $odor = f$ is also selected in the selection phase, covering 2160 positive and no negative examples.

The green path shows the top-down specialization of a rule using the $\overline{h}_{lap}$ heuristic. This heuristic prefers rules with high coverage of positive examples. It gives preference to rule refinements with the smallest angle between the line of the refinement and the horizontal axis, i.e. angles α, β, and γ in Fig. 5. Top-down specialization continues until there are no covered negative examples or there are no possible refinements. In Fig. 5 the refinement stops at point ***F***, where rule *veil-color* $=$ *w*, *gill-spacing* $=$ *c*, *bruises?* $=$ *f*, *ring-number* $=$ *o*, *stalk-surface-above-ring* $=$ *k* is constructed, covering a total number of 2192 positive examples and no negative examples. Using only a single selection heuristics this rule would be preferred to the rule depicted with the red path, but it is not achievable as a different choice was made already in the first step.

In summary, inverted heuristics prefer rules with high coverage of positive examples. The top-down specialization of a rule is steadily removing negative examples and some positive examples. This leaves the possibility that an additional refinement will construct a rule with the same or a higher number of covered positive examples than a rule constructed using a single heuristics which immediately maximizes its value.

2.5. Relation to previous work

Our work is closely related to previous work in rule learning and subgroup discovery. In particular, it explores the recommended approach by Stecher et al. (2014) for separation of rule refinement and rule selection and the use of different heuristics in the classification rule learning context. While rule induction algorithms and subgroup discovery algorithms typically use the same heuristic for rule refinement and rule selection, Stecher et al. (2014) argued that the nature of the separate-and-conquer algorithms offers the possibility of separating the two rule construction phases and their evaluation using two different heuristics.

Table 3
Description of coverage space points from Fig. 3, illustrated on the mushroom data set, using target class p (*poisonous*).

Point	Rule			p	n
U	p	←	true.	3916	4208
A	p	←	odor = f.	2160	0
B	p	←	veil-color = w.	3908	4016
C	p	←	veil-color = w, gill-spacing = c.	3804	2816
D	p	←	veil-color = w, gill-spacing = c, bruises? = f.	3188	160
E	p	←	veil-color = w, gill-spacing = c, bruises? = f, ring-number = o.	3152	144
F	p	←	veil-color = w, gill-spacing = c, bruises? = f, ring-number = o, stalk-surface-above-ring = k.	2192	0

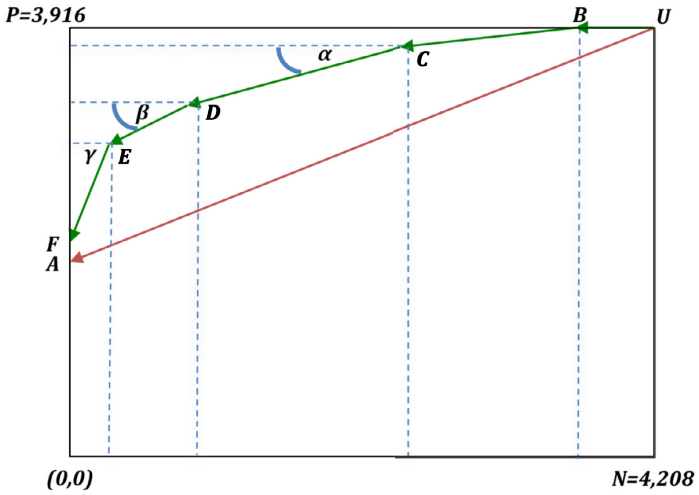


Fig. 5. Comparison of rule refinement paths using standard heuristic and the inverted one. The red path shows rule constructed using h_{lap}. The green path shows rule refinement using $\mathcal{U}_{lap}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

In this paper we investigate the separation of the rule refinement and rule selection phase in both subgroup discovery and classification rule learning. Along with the phase separation we introduce two beams, each consisting of the best rules according to the refinement and selection heuristic, respectively.

Contrary to the approach of Stecher et al. (2014), where they compare the selection quality of the best rule refinement to the rule with the best selection quality, we compare the selection quality of all refined candidate rules to the selection quality of the best rules for selection (current selection beam members). In this way we expand the space of possible candidates for selection and increase the possibility of choosing a candidate with good selection quality, which might have been omitted in the refinement phase using the Stecher et al. (2014) approach. Additionally, our algorithm for rule learning builds rule sets for each target class of a given data set. This is different from the approach taken in Stecher et al. (2014) where unordered decision lists are constructed.

This paper also significantly extends our previous work (Valmarska, Robnik-Šikonja, & Lavrač, 2015), where we reported on the initial findings regarding the use of inverted heuristics in subgroup discovery. In this paper, we introduce an additional heuristic, WRACC, which consequently proves to improve over other heuristics in several settings. In addition, we propose a different approach to algorithm comparison, by first determining the default parameters for each algorithm and then comparing the algorithms on new data sets, using the default parameters. The establishment of default parameters is valuable for future users of the algorithms, as it offers a solid starting point for their use. In addition to the subgroup discovery algorithm, in this paper we also introduce a novel classification rule learning algorithm based on double beam and compare it to the state-of-the-art rule learning algorithms.

3. DoubleBeam algorithm for subgroup discovery

The previously observed favorable properties of inverted heuristics in a classification setting provide a motivation to test the idea in the subgroup discovery context. For this purpose, we developed the DoubleBeam-SD subgroup discovery algorithm[1], which combines separate refinement and selection heuristics with the beam search. In the same fashion, we integrated the beam search and two separate heuristics in the classification rule learning setting, which we discuss in Section 4.

Contrary to conventional beam-search based algorithms such as CN2-SD (Lavrač et al., 2004), the DoubleBeam-SD algorithm for subgroup discovery maintains two separate beams, the *refinement beam* and the *selection beam*. Upon initialization, each beam is filled with the best single-condition rules according to their refinement and selection quality, respectively. The algorithm then enters a loop. In each iteration, rules of the form *targetClass* ← *Conditions* from the refinement beam are refined by adding features to the *Conditions* part of the existing rules. The resulting new rules are added to the refinement beam, which is ordered according to the refinement quality. Newly produced rules are then evaluated according to their selection heuristic and the selection beam is updated with the rules whose selection quality is better than the selection quality of the rules already stored in the beam. The algorithm exits the loop after the maximally allowed number of steps is reached. Another purpose of storing several rules in the selection beam is to allow post-processing where only the non-redundant subset of rules is retained (Gamberger & Lavrač, 2002). The DoubleBeam-SD algorithm is outlined in Algorithm 1.

In order to induce descriptions for subgroups of data instances which have not yet been covered by the previously constructed rules, we employ weighted covering, which reduces the weight of covered positive examples but does not remove them entirely. This required a modification of the method for updating the selection beam. Each time a positive example is covered by a rule that is already in the selection beam, the instance coverage count is increased and consequently the instance weight is decreased, which results in reducing the probability that the covered examples would be covered again by the rules constructed in the following iterations of the algorithm.

In this work, we used the harmonic and geometric weights for instance weighting. We also implemented removal of the already covered positive instances by assigning weight 0 to every instance already covered by some rule in the selection beam (method *zero weight*). Eqs. (10)–(12) show how the weight of a covered example is updated depending on the number of rules that cover it.

[1] Code is available on github at https://github.com/bib3rce/RL_SD.

Algorithm 1: DoubleBeam-SD algorithm.

Input: : $E = P \cup N$
E is the training set, $|E|$ its size,
tc is target class,
P are positive examples (of class tc),
N are negative examples (of classes $\neq tc$).
Output: : *subgroup descriptions*
Parameters: : *minSupport*,
rbw is refinement beam width,
sbw is selection beam width,
rh is refinement heuristic,
sh is selection heuristic
maxSteps is maximal number of steps

```
CandidateList ← all feature values or intervals
for each candidate in CandidateList do
    evaluate candidate with rh
    evaluate candidate with sh
end
sort CandidateList according to the rh
for i = 0 to rbw do
    RB[i] ← CandidateList[i]
end
sort CandidateList according to the sh
for i = 0 to sbw do
    SB[i] ← CandidateList[i]
end
step ← 1
do
    refinedCandidates ← refine RB with CandidateList
    replace RB with refinedCandidates using rh
    updateSelectionBeam(SB, refinedCandidates, sh)
    step ← step + 1
while step ≤ maxSteps;
return SB
```

Geometric weight:

$$w_g(d_i) = \alpha^k, \tag{10}$$

where k is the number of rules that have already covered example d_i;

Harmonic weight:

$$w_h(d_i) = \frac{1}{k+1}, \tag{11}$$

where k is the number of rules that have already covered example d_i;

Zero weight:

$$w_z(d_i) = 0, \tag{12}$$

if example d_i is covered by at least one rule in the selection beam.

The weighted value of positive examples covered by a rule r (*weighted number of true positives*) is calculated using Eq. (13).

$$wTP(r) = \sum_{i=1}^{|E|} w(d_i) \cdot c \quad \begin{cases} c = 1 & \text{if } r \text{ covers } d_i; \\ c = 0 & \text{otherwise.} \end{cases} \tag{13}$$

Note that zero weight can be understood as removing covered positive examples from the data set. This is not the same as no weighting, which means that instances are retained in the data set. As we use the selection beam, which keeps all the interesting subgroups, and the algorithm takes care that beam entries are not duplicated, no weighting might be sufficient. However, a practical reason to introduce instance weighting are possible redundancies in the attribute set. Without weighting we might get several different but redundant descriptions of the same instances in the beam, which unnecessary fill the beam and reduce the search space. The code of the **updateSelectionBeam** method is outlined in Algorithm 2.

Algorithm 2: Method for updating the selection beam.

```
Method updateSelectionBeam(SB, refinedCandidates, sh)
    // current data
    cData ← P ∪ N
    // candidates for selection
    cs ← ∪SB
    // new selection beam
    nSB ← {}
    resetWeights(cData)
    for i = 0 to sbw do
        bestRule ← getBestRule(cs, cData, sh)
        cs ← remove(cs, bestRule)
        nSB{i} ← bestRule
        cData ← updateWeights(cData, bestRule)
    end
    SB ← nSB
```

Function **getBestRule** returns the rule with the best selection quality on the data set with updated weights. The selection quality of a rule is calculated according to the chosen selection heuristics. Function **updateWeights** updates the weights of the covered positive examples. The weights are updated according to the desired weight type i.e. geometric, harmonic or zero.

3.1. Experimental setting

For the purpose of algorithm evaluation, we use different combinations of refinement and selection heuristics, constituting the following DoubleBeam subgroup discovery variants:

- **SD-ILL** (Inverted Laplace, Laplace), using $(\mathrm{U}_{lap}, \mathrm{h}_{lap})$ heuristics combination pair,
- **SD-IPP** (Inverted Precision, Precision), using $(\mathrm{U}_{prec}, \mathrm{h}_{prec})$,
- **SD-IMM** (Inverted M-estimate, M-estimate), using $(\mathrm{U}_{m\text{-}est}, \mathrm{h}_{m\text{-}est})$,
- **SD-IGG** (Inverted Generalization quotient, Generalization quotient), using $(\mathrm{U}_g, \mathrm{h}_g)$,
- **SD-GG** (Generalization quotient, Generalization quotient), using $(\mathrm{h}_g, \mathrm{h}_g)$, and
- **SD-WRACC** (WRACC), using $(\mathrm{h}_{\mathrm{WRACC}}, \mathrm{h}_{\mathrm{WRACC}})$.

For the purpose of annotation, we prefix the variants of our DoubleBeam-SD with SD. The h_g heuristic is the generalization quotient proposed in Gamberger and Lavrač (2002) (Eq. 4), while U_g is its inverted variant defined as $\mathrm{U}_g = \frac{N-n}{P-p+g}$. The weighted relative accuracy (WRACC) heuristic is defined in Eq. (5). It was introduced in Lavrač et al. (2004) to measure the unusualness of the induced subgroup describing rules. Note that WRACC is identical to its inverted variant (Stecher et al., 2014).

We compare three state-of-the-art subgroup discovery algorithms (SD, CN2-SD, and APRIORI-SD) and the proposed DoubleBeam-SD algorithm with six combinations of refinement and selection heuristics (SD-ILL, SD-IPP, SD-IMM, SD-IGG, SD-GG, and SD-WRACC). We test the DoubleBeam-SD algorithm with each of the six combinations of refinement and selection heuristics, both with and without using the weighted covering algorithm, and with and without using rule subset selection in the post-processing step described in Gamberger and Lavrač (2002). This resulted in 48 dif-

Table 4
Number of classes (C), examples (E), attributes (A), and features (F) of the 20 data sets used in the experiments.

Tuning data sets	C	E	A	F	Evaluation data set	C	E	A	F
breast-cancer	2	286	10	41	contact-lenses	3	24	5	9
car	4	1728	7	21	futebol	2	14	5	27
glass	7	214	10	31	ionosphere	2	351	35	157
hepatitis	2	155	20	41	iris	3	150	5	14
horse-colic	2	368	23	72	labor	2	57	17	42
hypothyroid	2	3163	26	60	mushroom	2	8124	23	116
idh	3	29	5	14	primary-tumor	22	339	18	37
lymphography	4	148	19	52	soybean	19	683	36	99
monk3	2	122	7	17	tic-tac-toe	2	958	10	27
vote	2	435	17	32	zoo	7	101	18	134

Table 5
Chosen variants of the DoubleBeam-SD algorithm. The overall rank is the rank of the algorithm among the 48 variants.

Heuristics combination	Overall rank	Average rank	Post-processing	Weight type
WRACC	1	7.30	No	None
IMM	2	7.45	Yes	None
GG	3	8.15	No	None
IGG	9	15.50	No	None
ILL	14	19.95	Yes	None
IPP	21	24.45	No	Zero

ferent combinations of the DoubleBeam-SD algorithm (6 refinement/selection combinations × 2 post-processing/no × 4 weighting/no).

We use SD, CN2-SD and APRIORI-SD implementations of algorithms that are available in the ClowdFlows platform (Kranjc, Podpečan, & Lavrač, 2012). We use the same 20 UCI classification data sets as Stecher et al. (2014) (see Table 4). In order to determine suitable settings, we randomly split the data sets into two groups: we use 10 randomly chosen data sets (shown on the left-hand side of Table 4) to determine default parameters of all competing methods, and the remaining 10 data sets (shown on the right-hand side of Table 4) to compare the best settings. The tuning of parameters is described in Section 3.2, while the methods comparison is presented in Section 3.3.

To compare the speed and scalability of the algorithms, we use the UCI *adult* data set which consists of 32,561 instances and 14 attributes. We do not use cross-validation on this data set but split it into training and test sets of different sizes.

3.2. Default parameter setting

We use the 10 left-hand side data sets from Table 4 for setting default parameters of the algorithms. The SD algorithm and the APRIORI-SD algorithm are both trained using rule subset selection in the post-processing step, as described in Gamberger and Lavrač (2002). Originally, the CN2-SD algorithm does not use rule selection in the post-processing.

The algorithms are initially tested with 10-fold double-loop cross-validation on each of the 10 data sets used for parameter tuning (named *tuning data sets* in the rest of this paper). For each algorithm (both the newly proposed algorithms as well as the existing algorithms SD, CN2-SD and APRIORI-SD), a grid of possible parameter values is set in advance. The value of *minSup* is set to 0.01. Each training set of a given cross-validation iteration is additionally split into an internal training and testing subset. For each algorithm, models were built using the internal training subset and the parameters from its own parameter grid. Parameters maximizing the value of unusualness of the produced subgroups on the internal test subest are then chosen for building a model using the whole training set. In the evaluation, we use the subgroup discovery evaluation statistics proposed in Kralj Novak et al. (2005) (originally implemented in the Orange data mining environment Demšar et al., 2013): *coverage, support, size, complexity, significance, unusualness* (WRACC), *classification accuracy*, and *AUC*.

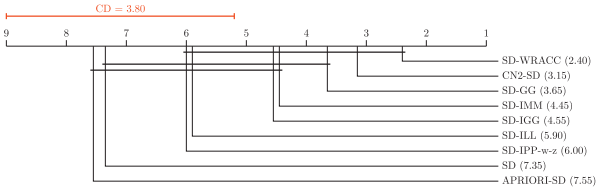


Fig. 6. Nemenyi test on ranking of subgroup discovery algorithms regarding average WRACC values with a significance level of 0.05.

We compute average ranks of the 48 combinations of the DoubleBeam-SD algorithm with respect to the unusualness (WRACC) of the produced subgroup describing rules. For each combination of refinement and selection heuristics of algorithms described in Section 3.1 we chose the algorithm setting that had the best average ranking. The chosen algorithm settings are shown in Table 5.

The default set of parameters for each algorithm consists of the parameters which were chosen in the 10-fold double-loop cross-validation testing phase. This default set of parameters is used for cross-validation testing of the subgroup discovery algorithms on the remaining 10 data sets.

3.3. Experimental results

The WRACC values obtained from the 10-fold cross-validation testing on the 10 evaluation data sets with selected default parameters are shown in Table 6. These values are averaged over all the classes for every particular data set.

The results of the Nemenyi test following the Friedman test for statistical significance of differences between average values of WRACC are shown in Fig. 6. It is evident that SD-WRACC algorithm produces the most interesting subgroups, which are statistically more unusual than the ones produced by the two state-of-the-art algorithms, the SD algorithm and the APRIORI-SD algorithm. However, there are no statistically significant differences between the

Table 6
Ten-fold cross-validation WRACC results for subgroup discovery algorithms with default parameters. The best values for each data set are written in bold. We compare existing SD, CN2-SD and APRIORI-SD algorithms with the proposed DoubleBeam algorithms with different refinement and selection heuristics.

Data sets	SD	CN2-SD	APRIORI-SD	SD-ILL	SD-IPP-w-z	SD-IMM	SD-WRACC	SD-GG	SD-IGG
contact-lenses	0.032	0.071	0.027	0.039	0.035	0.021	0.047	**0.081**	**0.081**
futebol	0.000	0.009	0.005	0.005	0.003	**0.015**	0.000	0.006	0.005
ionosphere	0.099	0.111	0.000	0.083	0.032	0.105	**0.133**	0.105	0.107
iris	0.090	**0.200**	0.142	0.159	0.167	0.146	0.175	0.148	0.148
labor	0.080	**0.102**	0.041	0.081	0.085	0.085	0.098	0.095	0.094
mushroom	0.088	0.163	0.000	0.133	0.029	0.134	**0.191**	0.146	0.131
primary-tumor	0.011	0.009	0.008	0.006	0.006	0.017	**0.019**	0.014	0.014
soybean	0.025	0.037	0.000	0.035		0.036	**0.043**	0.037	0.035
tic-tac-toe	0.022	0.021	0.029	0.024	0.029	0.024	**0.041**	0.028	0.029
zoo	0.037	0.097	0.000	0.094	0.065	0.096	**0.100**	0.099	0.094

Table 7
Performance comparison of subgroup discovery algorithms using WRACC score and average rule length (ARL) on the UCI *adult* data set. The data set is split in 70:30 ratio. Rules are induced using the default parameters.

Measure	SD	CN2-SD	APRIORI-SD	SD-ILL	SD-IPP-w-z	SD-IMM	SD-WRACC	SD-GG	SD-IGG
WRACC	0.023	0.043	0.041	0.011	0.012	0.028	0.076	0.025	0.024
ARL	2.800	2.150	2.700	2.800	1.300	2.100	2.100	2.600	2.500

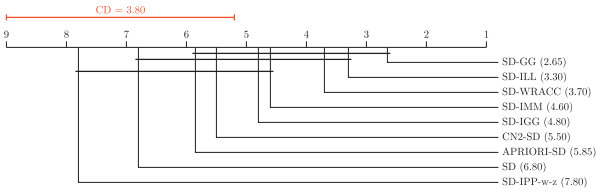


Fig. 7. Nemenyi test on ranking of average rule sizes for subgroup discovery algorithms in the second experimental setting with a significance level of 0.05. Note that algorithms are ordered according to the average length of generated rules—rank 1 would indicate the algorithm producing the longest rules.

six chosen variants of the DoubleBeam-SD algorithm and the CN2-SD algorithm. The DoubleBeam-SD algorithm with the combination (h_g, h_g) produces statistically more unusual subgroups than the ones produced by the APRIORI-SD algorithm. The rest of the variants of the DoubleBeam-SD algorithm do not produce subgroup describing rules which are statistically more interesting than the ones produced by any of the tested algorithms.

Experimental results reveal that algorithms which use WRACC as their heuristic (the SD-WRACC algorithm and the CN2-SD algorithm) produce rules which describe more interesting subgroups. The underperformance of the other considered variants of the DoubleBeam-SD algorithm is due to their respective heuristics, which are specialized towards finding prediction rules and not unusual rules.

The results of the Nemenyi test following Friedman test for statistical significance of differences of the average rule sizes are shown in Fig. 7. The DoubleBeam-SD algorithm with the combination (h_g, h_g) produces subgroups which are on average described by the longest rules. The SD-GG algorithm generates subgroups described by rules that are statistically longer only than the ones produced by the SD algorithm and the SD-IPP algorithm with zero-weight covering. There is no statistical evidence that the SD-GG algorithm produces longer rules than other evaluated algorithms. Consequently, these results do not confirm that the DoubleBeam-SD algorithm with inverted refinement heuristic produces statistically longer subgroup descriptions than all other subgroup discovery algorithms. This is slightly surprising taking into account the findings of Stecher et al. (2014) in the classification rule learning setting.

Table 7 presents the performance of subgroup discovery algorithms on the *adult* data set in terms of their WRACC score. We split the data set in the 70:30 ratio, leading to 22,793 training and 9768 testing instances. The SD-WRACC algorithm produced the most interesting rules, followed by the CN2-SD algorithm. The results are in accordance with the results presented in Fig. 6. The algorithms SD-ILL, SD, APRIORI-SD and SD-GG generated the longest rules; the SD-ILL and SD-GG algorithms produced the longest rules also on data sets from Fig. 7.

Fig. 8 presents the *training* times of subgroup discovery algorithms with different numbers of training instances from the *adult* data set. The APRIORI-SD algorithm is the slowest, followed by the SD-IPP-w-z and CN2-SD algorithms. The other subgroup discovery algorithms are comparable in terms of training time and allow for processing of relatively large data sets.

To the users of subgroup discovery algorithms we recommend the use of the SD-WRACC algorithm with the selection beam width set to 5, and no example weighting or post-processing. Results show that this algorithm on average outperforms other subgroup discovery algorithms considered in this work.

4. DoubleBeam algorithm in classification rule learning

The idea of using two separate heuristics for rule refinement and selection as well as using inverted heuristics in refinement phase was proposed and successfully tested by Stecher et al. (2014). The previous section shows that this idea can also be successful in subgroup discovery, where we tested it using a double beam search approach. As Stecher et al. (2014) do not use beam search in rule learning, an obvious extension is to use double beam also in classification rule learning.

In order to test the influence of different selection heuristics, refinement heuristics, selection beam width, and refinement beam width, we implemented a DoubleBeam classification rule learning (DoubleBeam-RL) algorithm. This algorithm is adaptation of the DoubleBeam-SD algorithm. It uses a combination of refinement and selection heuristics for each phase of rule learning. The algorithm has two beams, the selection beam and the refinement beam, where during the process of generating rules it holds potential candidates for refinement and selection, based on their selection and refinement quality. For learning a decision list, it employs the commonly used separate-and-conquer strategy (Fürnkranz, 1999): each time a rule is generated for a given target

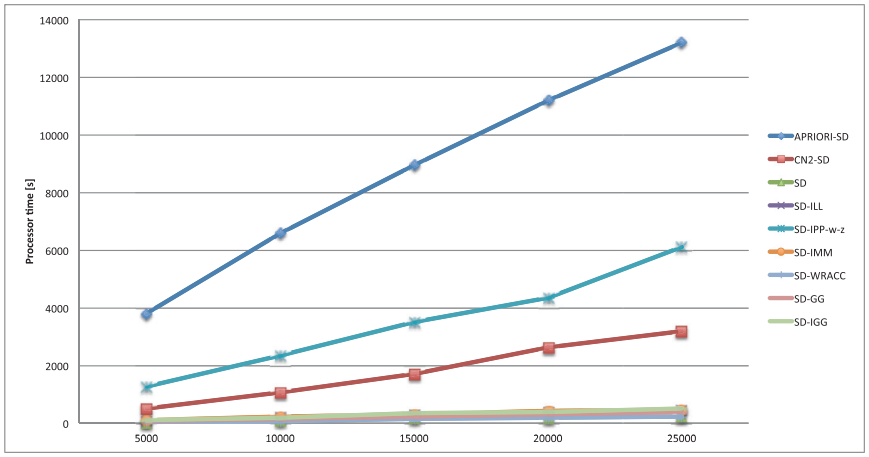


Fig. 8. Comparison of training times for subgroup discovery on the *adult* data set. The horizontal axis shows the number of training instances and the vertical axis shows the training time in seconds.

class, the positive examples covered by the rule are removed from the data set. The algorithm continues to learn new rules for the same target class on the updated data set as long as rules with a minimal acceptable quality are induced, i.e. if the rule covers more positive than negative examples and covers more positive examples than a chosen threshold (in our case a threshold of 2). The final result is a rule set with acceptable rules for the given target class.

Basically, for learning a single rule, a single beam (in the refinement phase) is sufficient, unless we want to produce a collection of rules which are post-processed later. If not, we shall set the selection beam width to 1, as we do in our experiments. The DoubleBeam-RL algorithm is outlined in Algorithm 3. The function

Algorithm 3: DoubleBeam-RL algorithm.

Input: : $E = P \cup N$
E is the training set, $|E|$ its size,
tc is target class,
P are positive examples (of class tc),
N are negative examples (of classes $\neq tc$).
Output: : R, (R is rule set for tc)
Parameters: : rh is refinement heuristic,
sh is selection heuristic,
rbw is refinement beam width,
sbw is selection beam width.

```
// rule set for target class tc
R ← {}
// current data
cData ← E
do
    rule ← generateRule(cData, tc, rh, sh, rbw, sbw)
    R ← R + rule
    cData ← removePositiveCovered(cData, rule, tc)
while not satisfied;
return R
```

for generating a single rule when a data set, selection heuristics, refinement heuristics, selection beam width, and refinement beam width are given is outlined in Algorithm 4.

4.1. Experimental setting

We perform experimental evaluation in two steps. In the first step we determine default parameters for the five best combinations of refinement and selection heuristics on the same randomly chosen 10 data sets in the left-hand side of Table 4. In the second step, we use 10 fresh data sets (the right-hand side of Table 4) to compare these five best configurations with two state-of-the-art algorithms for rule learning, Ripper (Cohen, 1995) and CN2 (Clark & Niblett, 1989). We use the Weka (Hall et al., 2009) implementation of Ripper and the Orange (Demšar et al., 2013) implementation of the CN2 algorithm. For both algorithms we use the default parameters set by their software platforms, respectively. For comparison, we also include the results from the best performing algorithm from Stecher's (Stecher et al., 2014) experimental work, named SC-ILL.

The quality of the induced rules is measured in terms of the classification accuracy (CA). The process of parameter tuning and variant selection is described in Section 4.3. We also report the average rule length of produced rules.

Algorithm 4: Function for generating rules using two heuristics.

```
Function generateRule (dataset, tc, rh, sh, rbw, sbw)
    // candidates for best rule
    bRC ← DoubleBeam-SD(dataset, tc, rh, sh, rbw, sbw)
    bestRule ← getBestRule(bRC)
    return bestRule
```

4.2. Illustrative example

We compare our approach with the approach of Stecher et al. (2014) with an illustrative example. For the purpose of this comparison, we chose the same set of attributes used in the mentioned work. Rules in both decision lists are generated with H_{lap} as the refinement heuristic and h_{lap} as the selection heuristic. The width of both refinement and selection beam is set to 1. Fig. 8 shows the decision list learned for the class *poisonous* on the data set *mushroom* using the algorithm presented in Stecher et al. (2014), whereas Fig. 9 presents the rule set learned by our DoubleBeam rule learning algorithm.

Results from Tables 8 and 9 suggest that our approach tends towards finding even more complete rules than the approach taken by Stecher et al. (2014). The algorithm produces

Table 8
Decision list learned for class p (*poisonous*) in the mushroom data set using Stecher's approach with refinement heuristic $\mathcal{U}_{lap}$ and selection heuristic h_{lap}. The number of positive examples covered by each rule is also shown. No rule covers any of the negative examples.

2192	p	←	veil-color = w, gill-spacing = c, bruises? = f, ring-number = o, stalk-surface-above-ring = k.
864	p	←	veil-color = w, gill-spacing = c, gill-size = n, population = v, stalk-shape = t.
336	p	←	stalk-color-below-ring = w, ring-type = p, stalk-color-above-ring = w, ring-number = o, cap-surface = s, stalk-root = b, gill-spacing = c.
264	p	←	stalk-surface-below-ring = s, stalk-surface-above-ring = s, ring-type = p, stalk-shape = e, veil-color = w, gill-size = n, bruises? = t.
144	p	←	stalk-shape = e, stalk-root = b, stalk-color-below-ring = w, ring-number = o.
72	p	←	stalk-shape = e, gill-spacing = c, veil-color = w, gill-size = b, spore-print-color = r.
44	p	←	stalk-surface-below-ring = y, stalk-root = c.

Table 9
Rule set learned for the class p (*poisonous*) in the mushroom data set using DoubleBeam rule learning algorithm with refinement heuristic $\mathcal{U}_{lap}$, selection heuristic h_{lap}, and both refinement and selection beam width set to 1. The number of positive examples covered by each rule is shown on the left. No rule covers any negative examples.

2228	p	←	p ← gill-spacing = c, veil-color = w, stalk-surface-above-ring = k.
864	p	←	gill-color = b.
336	p	←	stalk-color-above-ring = w, gill-spacing = c, stalk-root = b, stalk-color-below-ring = w, gill-attachment = f, cap-surface = s, ring-number = o, ring-type = p.
264	p	←	stalk-shape = e, bruises? = t, gill-size = n, gill-attachment = f, stalk-surface-above-ring = s, stalk-surface-below-ring = s, ring-type = p.
144	p	←	stalk-shape = e, bruises? = f, stalk-root = b, stalk-color-below-ring = w, gill-attachment = f.
72	p	←	stalk-shape = e, spore-print-color = r.
8	p	←	veil-color = y.

on average shorter rules which include more or the same number of examples. The DoubleBeam-RL algorithm is able to detect features that do not contribute to the overall improvement of the rules. Such example is the bruises? = f feature. In the first rule from Stecher's decision rule, the 2192 covered examples are covered by the conjunction of the following features: veil-color = w, gill-spacing = c, ring-number = o, stalk-surface-above-ring = k. Feature bruises? = f was selected during the refinement phase, but does not contribute anything to the final result.

This difference is due to the nature of the applied algorithms. Stecher's approach is to refine a rule using the inverted heuristics until there are only positive examples covered and then returns the best rule on the refinement path. This approach leads to eliminating possible refinements of a certain rule due to their lower refinement quality, even though their selection quality is very high; in our case, one of the possible refinements has even better selection quality than the final rule, chosen by the Stecher's approach. The DoubleBeam-RL algorithm on the other hand, considers the selection quality of the refined candidates and the rules already in the selection beam. It simultaneously checks for rules with best refinement and selection quality and keeps track of all the best rules found in the refinement process.

As an example, consider rules from Table 3. After the universal rule is refined, the best candidate for further refinement in both approaches is p ← veil-color = w. The DoubleBeam-RL algorithm saves this rule as a candidate for refinement, but chooses rule p ← odor = f as its candidate for best rule. In the next iteration, once more the two algorithms have the best candidate for refinement, p ← veil-color = w, gill-spacing = c. There is no change in the selection beam of the DoubleBeam-RL algorithm, where the selection quality of p ← odor = f (1.000) is better than the selection quality of p ← veil-color = w, gill-spacing = c (0.575). In the third step, best rule for refinement is p ← veil-color = w, gill-spacing = c, bruises? = f. Both algorithms will continue with the refinement of this rule, however, the selection beam of the DoubleBeam-RL algorithm will be updated with a refinement of p ← veil-color = w, gill-spacing = c, leading to rule p ← veil-color = w, gill-spacing = c, stalk-surface-above-ring = k, whose selection quality is the same as the selection quality of the rule already stored in the beam (1.000). When the DoubleBeam-RL algorithm is faced with choosing between two rules with the same selection quality, it always chooses the rule that has covered more positive examples. In case the decision is not straight-forward, it chooses the shortest among the rules in question. The top-down specialization will continue for both algorithms. The algorithm proposed by Stecher will stop when there are only positive examples covered or there is no possible further refinement. At the end, the algorithm will return the rule with the best selection quality among all the rules on the refinement path. As it is evident from our example, this will result with longer rules which can have lower coverage than the rules selected by the DoubleBeam-RL algorithm. The DoubleBeam-RL algorithm stops after a predefined number of steps, and returns the rule with the best selection quality among all the investigated refinements.

4.3. Default parameter setting

In the experiments performed to determine the default parameter values for the DoubleBeam-RL algorithm, we use all combinations of the following heuristics in refinement and selection phase:

- Laplace h_{lap} - L,
- Inverted Laplace $\mathcal{U}_{lap}$ - IL,
- Precision h_{prec} - P,
- Inverted Precision $\mathcal{U}_{prec}$ - IP,
- M-estimate $h_{m\text{-}est}$ - M,
- Inverted -M-estimate $\mathcal{U}_{m\text{-}est}$ - IM, and

– Weighted Relative ACCuracy h_{WRACC} - W.

As an example, the abbreviation RL-ILL indicates that $^{q}l_{ap}$ was used as a refinement heuristic, and h_{lap} as a selection heuristic in the DoubleBeam-RL algorithm. This resulted in 49 variants of the DoubleBeam-RL algorithm. Each variant is tested on the same 10 randomly chosen data sets that were used for parameter tuning in the subgroup discovery context.

The value of the selection beam width is fixed to 1 in all variants (see Section 4). In order to select the default width of the refinement beam for each variant of the DoubleBeam-RL algorithm, we perform a 10-fold double-loop cross-validation of each variant on each of the tuning data sets from Table 4. Each tuning data set is divided into training and test set. Each training data set is additionally split into internal training and test subset. A separate model is induced on the internal training subset for each of the possible parameter values. These models are then evaluated using the internal test subset. The parameter values that maximize the value of classification accuracy are chosen as parameters for the construction of the model using the initial training data set. The final cross-validation value of classification accuracy (CA) for each fold is calculated using the model induced with the chosen best parameters and the corresponding test data set.

For each heuristic combination we collected the best parameters for refinement beam width across the tested 10 data sets. The most frequently selected parameter was chosen as a default parameter for the considered combination. Our experiments showed that for each variant of the rule learning algorithm, the most accurate rules are induced when we use refinement beam width with value 1. This means that the selected best parameters make our algorithm identical to the rule learning algorithm proposed in Stecher et al. (2014), with the exception that our algorithms can select the best rule from each refinement step (line 2 of Algorithm 4 and line 18 of Algorithm 1), while in Stecher et al. (2014) only the final refined rule is selected. This seemingly small difference leads our algorithm to form shorter rules with better coverage and affects also the classification accuracy as presented in Section 4.4.

Out of the 49 variants of the DoubleBeam-RL algorithm, we eventually selected the following five variants, which had the best average rank performance on the 10 tuning data sets: RL-MM ($h_{m\text{-}est}$, $h_{m\text{-}est}$), RL-ILM ($^{q}l_{ap}$,$^{h}m\text{-}est$), RL-WM (h_{WRACC}, $h_{m\text{-}est}$), RL-IPM ($^{q}l_{prec}$,$^{h}m\text{-}est$), and RL-PM (h_{prec}, $h_{m\text{-}est}$).

4.4. Experimental results

We compare the selected best rule learning algorithm (using the five chosen variants of rule selection heuristics) with two state-of-the art algorithms, Ripper and CN2, and the best performing algorithm from Stecher et al. (2014)'s work, named SC-ILL. The classification accuracy (CA) values obtained from the 10-fold cross-validation testing on the 10 evaluation data sets from the right-hand side of Table 4 with default parameters are shown in Table 10.

The Friedman test for statistical differences in CA showed that there are no significant differences between the algorithms which is confirmed by the confidence intervals of the Nemenyi test in Fig. 9. Nevertheless, the approach taken by Stecher et al. (2014) yields the best results (average rank is 3.00). Three of our five chosen variants of the DoubleBeam-RL algorithm have a better average rank than the Ripper algorithm. The chosen variants of our algorithm for rule learning on average perform better than the CN2 algorithm.

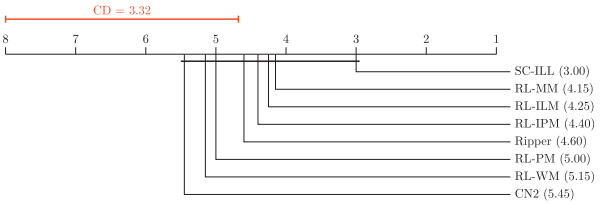


Fig. 9. Nemenyi test on ranking of classification accuracy values with a significance level of 0.05.

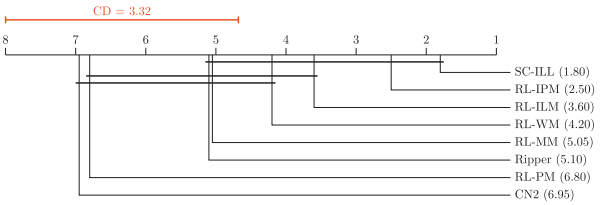


Fig. 10. Nemenyi test on ranking of average classification rule length with a significance level of 0.05.

An interesting observation is that among all the algorithms with two heuristics, the one with the least search performs best i.e. the Stecher et al. (2014) approach. The explanation for this could be the *over-searching* phenomenon (Janssen & Fürnkranz, 2009; Quinlan & Cameron-Jones, 1995), which indicates that the amount of search shall be adjusted specifically to a data set and search heuristics employed.

The results of the Friedman test and post-hoc Nemenyi test for statistical significance of differences between average rule length of rules induced by the chosen variants of the DoubleBeam-RL algorithm and the state-of-the-art algorithms for classification rule learning are shown in Fig. 10. The results suggest that the variant that uses the inverted heuristic in refinement phase, RL-IPM, induces rules which are statistically longer than the rules induced by the standard refinement heuristic, RL-PM. The results in Fig. 10 are in accordance with the conclusions drawn by Stecher et al. (2014). The approach taken by Stecher et al. (2014), SC-ILL, produces longest rules, while the CN2 algorithm produces rules with the shortest average rule length. These rules are significantly shorer than the rules produced by the SC-ILL, the Rl-IPM, and the RL-ILM algorithm. Note the average rule length is calculated as the ratio between the sum of all conditions across all induced rules and the total number of rules in the model.

Table 11 shows the performance comparison of classification rule learning algorithms on the *adult* data set in terms of classification accuracy and average rule length. Results reveal that all versions of the DoubleBeam-RL algorithm produce rules with better classification accuracy than the CN2 algorithm. Three DoubleBeam-RL algorithms (RL-ILM, RL-WM, and RL-IPM) slightly outperform the Ripper algorithm. Comparison of the results obtained with the algorithms RL-IPM and RL-PM confirm the conclusions of Stecher et al. (2014): when an inverted heuristic is used in the refinement phase, the produced rules tend to be longer and have better classification accuracy.

Fig. 11 presents the *training* times of classification rule learning algorithms with different numbers of training instances from the *adult* data set. The times can only give a rough picture of the algorithms' performance, as the algorithms are not implemented on the same platform: we use the Ripper implementation from Weka, CN2 from Orange, the other algorithms are implemented in Python. The training times of the SC-ILL algorithm are not included due to excessive time consumption of the algo-

Table 10
Ten-fold cross-validation CA results for rule learning with default parameters. Best values are written in bold.

Data sets	RL-MM	RL-ILM	RL-WM	RL-IPM	RL-PM	Ripper	SC-ILL	CN2
contact-lenses	0.750	0.750	0.750	0.750	0.750	0.750	**0.875**	0.683
futebol	0.700	0.700	0.700	0.700	0.700	0.571	0.571	**0.800**
ionosphere	0.900	0.861	0.858	0.875	0.914	0.897	**0.932**	0.906
iris	0.920	0.920	0.920	0.920	0.920	**0.953**	**0.953**	0.893
labor	0.773	0.820	0.720	0.820	**0.827**	0.771	0.825	0.720
mushroom	0.999	**1.000**	**1.000**	**1.000**	0.997	**1.000**	**1.000**	**1.000**
primary-tumor	0.401	0.407	**0.410**	0.395	0.345	0.392	0.360	0.345
soybean	0.921	0.908	0.903	0.909	0.852	0.915	**0.924**	0.883
tic-tac-toe	**0.982**	0.976	0.892	0.974	0.980	0.978	0.976	0.818
zoo	0.872	0.892	0.882	0.892	0.823	0.871	0.921	**0.961**

Table 11
Comparison of classification accuracy (CA) and average rule length (ARL) of classification rule learning algorithms on the UCI *adult* data set. Data set is split in 70:30 ratio. Models are induced using estimated default parameters. SC-ILL results are not included as its training took more than 5 hours of CPU time.

Measure	RL-MM	RL-ILM	RL-WM	RL-IPM	RL-PM	Ripper	SC-ILL	CN2
CA	0.834	0.851	0.852	0.854	0.835	0.845	/	0.815
ARL	2.909	2.824	1.938	2.684	1.214	4.333	/	2.531

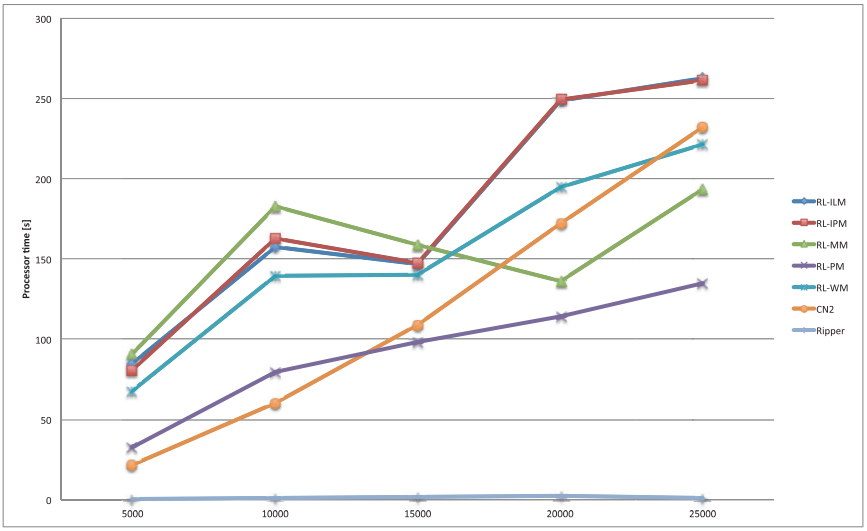


Fig. 11. Comparison of training times for classification rule learning algorithms on the *adult* data set. The horizontal axis shows the number of training instances and the vertical axis shows the training time in seconds.

rithm. Fig. 11 shows that the Ripper algorithm is the fastest classification rule learner. Algorithms RL-IPM and RL-ILM have almost identical training times and are the most inefficient. An interesting observation is that DoubleBeam-RL algorithms, which use inverted heuristics in their refinement phase, produce slightly more accurate models (Table 11) than their Laplace counterparts at the cost of being less efficient. Fig. 11 reveals that algorithms RL-ILM, RL-IPM, RL-MM, and RL-WM may use less time in spite of larger training set. Further investigation revealed relatively large variance of measured times. For specific points the mentioned algorithms produce models with fewer rules and fewer conditions.

Based on our experimental work, there can be no clear recommendation for the user which algorithm to use, as the differences in classification accuracy are not statistically significant. However, several algorithms with two heuristics produces on average more accurate rules than Ripper and CN2, the most accurate being SC-ILL and RL-MM. For large data sets where computational efficiency is crucial, Ripper is clearly the best choice.

5. Conclusions

This paper introduces two new algorithms for rule learning, one for subgroup discovery and one for classification rule learning. Both algorithms use beam search and offer the possibility to use separate heuristics for rule refinement and rule selection.

The experiments were performed on 20 UCI data sets. The performance of each of the considered algorithms depends on its parameters. In order to systematically choose the default parameters for each algorithm, we initially performed 10-fold double-loop cross-validation training on 10 randomly chosen data sets. The conclusions about the performance of the discussed algorithms are obtained after ten-fold cross-validation testing on the remaining 10 data sets, which have not been used for parameter setting and are exclusively used for algorithm evaluation.

The experiments indicate that the subgroup describing rules created using the SD-WRACC algorithm are more interesting than the subgroups induced by other state-of-the-art subgroup discov-

ery algorithms. The difference between most of the algorithms are not statistically significant, however SD-WRACC and CN2-SD produce statistically significantly more interesting rules than SD and APRIORI-SD.

In the context of classification rule learning we proposed a new, the DoubleBeam-RL algorithm, which offers the possibility for using separate rule refinement and selection heuristics. Among the tested 49 variants of refinement and selection heuristics inside the DoubleBeam-RL algorithm and their comparison with Ripper and CN2, the best performing variants in terms of classification accuracy were the algorithms that use the m-estimate as their selection heuristic. In particular, the best performing variant of the DoubleBeam-RL algorithm was the variant that uses the m-estimate both as its selection and refinement heuristic. The differences are, however, not statistically significant. The algorithms which use inverted heuristic perform slightly better than the algorithms using the standard heuristics (RL-IPM and RL-ILM compared to RL-PM and RL-WM). All five of our algorithms perform better than the CN2 algorithm, and three of our algorithms perform better than the Ripper algorithm.

The main advantage of DoubleBeam-SD and DoubleBeam-RL algorithms is their ability to use separate heuristics for the refinement and selection phase of rule learning. Different heuristics can take advantage of the data properties and contribute to better rules (rules with improved unusualness or rules with improved accuracy). The experimental results suggest that both algorithms provide rules with comparable or better quality than those obtained by the state-of-the-art algorithms for rule learning and subgroup discovery, respectively. The use of two beams in combination with separate heuristics for each phase of the learning processes widens the algorithms' search space thus improving the probability of finding better quality rules. However, this also increases the chances of data overfitting, an aspect which our algorithms do not explicitly address at this point.

In contrast to the APRIORI-SD algorithm which uses exhaustive search, the DoubleBeam-SD algorithm is a heuristic search algorithm (similar to the SD and the CN2-SD algorithm). Despite being faster than the APRIORI-SD algorithm and ability to handle medium size data sets, the current DoubleBeam-SD algorithm is still not able to handle large data sets, due to space and time complexity. In fact, this is one of the main disadvantages of all rule learning algorithms using a covering approach. Lower memory consumption could be achieved with more efficient data structures, while significant speedups could be gained with instance sampling and feature subset selection, as well as with parallelization of the algorithms. Due to two beams, large degree of parallelization could be achieved with DoubleBeam algorithms.

While our DoubleBeam-SD and DoubleBeam-RL algorithms show promising results, their increased search power demands further research in terms of stopping criteria and rule pruning heuristics. Using a post-processing rule pruning step similar to the Ripper is a promising research direction. We plan to explore also the rule pruning method proposed by Sikora (2011).

Experimental results on subgroup discovery revealed the advantage of using WRACC over the traditional rule learning heuristics in obtaining interesting subgroups. We believe that developing new heuristics specialized for the detection of interesting subgroups is a promising research path.

Subgroup discovery is a useful approach in the analysis of medical data. In line with our work on Parkinson's disease data (Valmarska, Miljkovic, Robnik-Šikonja, & Lavrač, 2016), we plan a case-study comparing results of different subgroup discovery algorithms on Parkinson's disease patients data set. In order to increase the interpretability of the induced subgroup describing rules we also plan on presenting a method for subgroup visualization. In this way we will assist experts (e.g. physicians) in their decision whether a certain subgroup discovery rule is interesting and relevant for their work.

Acknowledgments

The authors acknowledge the financial support from the Slovenian Research Agency (research core fundings No. P2-0209 and P2-0103). This research has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No. 720270 (HBP SGA1). We would like to thank Julius Stecher for interesting discussions on inverted heuristics.

References

Adhikari, P. R., Vavpetič, A., Kralj, J., Lavrač, N., & Hollmén, J. (2014). Explaining mixture models through semantic pattern mining and banded matrix visualization. In *Discovery science - 17th international conference, DS 2014, Bled, Slovenia, october 8–10, 2014. proceedings* (pp. 1–12).

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Vldb'94, Proceedings of 20th international conference on very large data bases, september 12–15, 1994, Santiago de Chile, Chile* (pp. 487–499).

Atzmueller, M. (2015). Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 5*(1), 35–49.

Atzmüller, M., & Puppe, F. (2006). SD-map - A fast algorithm for exhaustive subgroup discovery. In *Proceedings of knowledge discovery in databases, PKDD 2006* (pp. 6–17).

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

Bibal, A., & Frénay, B. (2016). Interpretability of machine learning models and representations: an introduction. In *Computational intelligence and machine learning, proceedings of European symposium on artificial neural networks ESANN 2016* (pp. 77–82).

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning, 3*, 261–283.

Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the twelfth international conference on machine learning* (pp. 115–123).

Demšar, J., Curk, T., Erjavec, A., Črt Gorup, Hočevar, T., Milutinovič, M., … Zupan, B. (2013). Orange: Data mining toolbox in Python. *Journal of Machine Learning Research, 14*, 2349–2353.

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, 1999* (pp. 43–52).

Fürnkranz, J. (1999). Separate-and-conquer rule learning. *Artificial Intelligence Review, 13*(1), 3–54.

Fürnkranz, J., & Flach, P. A. (2005). ROC 'n' rule learning - Towards a better understanding of covering algorithms. *Machine Learning, 58*(1), 39–77.

Fürnkranz, J., Gamberger, D., & Lavrač, N. (2012). *Foundations of rule learning*. Springer.

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

Gamberger, D., & Lavrač, N. (2000). Confirmation rule sets. In *Proceedings of principles of data mining and knowledge discovery, 4th European conference, PKDD 2000* (pp. 34–43).

Grosskreutz, H., & Rüping, S. (2009). On subgroup discovery in numerical domains. *Data Mining and Knowledge Discovery, 19*(2), 210–226.

Hall, M. A., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations, 11*(1), 10–18.

Hühn, J., & Hüllermeier, E. (2009). Furia: An algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery, 19*(3), 293–319.

Janssen, F., & Fürnkranz, J. (2009). A re-evaluation of the over-searching phenomenon in inductive rule learning.. In *Proceedings of the 2009 SIAM international conference on data mining* (pp. 329–340). SIAM.

Kavšek, B., Lavrač, N., & Jovanoski, V. (2003). APRIORI-SD: adapting association rule learning to subgroup discovery. In *Advances in intelligent data analysis v, 5th international symposium on intelligent data analysis, IDA* (pp. 230–241).

Klösgen, W. (1996). Explora: A multipattern and multistrategy discovery assistant. In *Advances in knowledge discovery and data mining* (pp. 249–271).

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

Kralj Novak, P., Lavrač, N., Zupan, B., & Gamberger, D. (2005). Experimental comparison of three subgroup discovery algorithms: Analysing brain ischemia data. In *Proceedings of the 8th international multiconference information society* (pp. 220–223).

Kranjc, J., Podpečan, V., & Lavrač, N. (2012). ClowdFlows: A cloud based scientific workflow platform. In *Proceedings of machine learning and knowledge discovery in databases - European conference, ECML PKDD 2012* (pp. 816–819).

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

Lavrač, N., Železný, F., & Flach, P. A. (2002). RSD: Relational subgroup discovery through first-order feature construction. In *Proceedings of the 12th international conference on inductive logic programming* (pp. 149–165).

Lichman, M. (2013). UCI machine learning repository.
Michalski, R. S. (1969). On the quasi-minimal solution of the general covering problem. In *Proceedings of the fifth international symposium on information processing*.
Minnaert, B., Martens, D., De Backer, M., & Baesens, B. (2015). To tune or not to tune: Rule evaluation for metaheuristic-based sequential covering algorithms. *Data Mining and Knowledge Discovery, 29*(1), 237–272.
Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence, 18*(2), 203–226.
Napierala, K., & Stefanowski, J. (2015). Addressing imbalanced data with argument based rule learning. *Expert Systems with Applications, 42*(24), 9468–9481.
Parpinelli, R. S., Lopes, H. S., & Freitas, A. A. (2002). Data mining with an ant colony optimization algorithm. *IEEE Transactions on Evolutionary Computation, 6*(4), 321–332.
Pičulin, M., & Robnik-Šikonja, M. (2014). Handling numeric attributes with ant colony based classifier for medical decision making. *Expert Systems with Applications, 41*(16), 7524–7535.
Quinlan, J., & Cameron-Jones, R. (1995). Oversearching and layered search in empirical learning. In *Proceedings of the 14th international joint conference on artificial intelligence, IJCAI'95: 2* (pp. 1019–1024). Morgan Kaufmann Publishers Inc.
Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess end games. *Machine Learning*.
Quinlan, J. R., & Cameron-Jones, R. M. (1993). FOIL: A midterm report. In *Proceedings of machine learning: European conference on machine learning - ECML 1993* (pp. 3–20).
Ruz, G. A. (2016). Improving the performance of inductive learning classifiers through the presentation order of the training patterns. *Expert Systems with Applications, 58*, 1–9.
Sikora, M. (2011). Induction and pruning of classification rules for prediction of microseismic hazards in coal mines. *Expert Systems with Applications, 38*(6), 6748–6758.
Stecher, J., Janssen, F., & Fürnkranz, J. (2014). Separating rule refinement and rule selection heuristics in inductive rule learning. In *Proceedings of machine learning and knowledge discovery in databases - European conference, ECML PKDD 2014* (pp. 114–129).
Valmarska, A., Miljkovic, D., Robnik-Šikonja, M., & Lavrač, N. (2016). Multi-view approach to Parkinson's disease quality of life data analysis. *Lecture Notes in Computer Science*. Springer.
Valmarska, A., Robnik-Šikonja, M., & Lavrač, N. (2015). Inverted heuristics in subgroup discovery. In *Proceedings of the 18th international multiconference information society*.
Vavpetič, A., Novak, P. K., Grčar, M., Mozetič, I., & Lavrač, N. (2013). Semantic data mining of financial news articles. In *Proceedings of the 16th international conference discovery science, DS 2013* (pp. 294–307).
Wrobel, S. (1997). An algorithm for multi-relational discovery of subgroups. In *Proceedings of the first European symposium on principles of data mining and knowledge discovery, PKDD 1997* (pp. 78–87).
Zeng, Q., Patel, J. M., & Page, D. (2014). Quickfoil: Scalable inductive logic programming. *Proceedings of Very Large Databases Conference, 8*(3), 197–208.

Chapter 5

Detection of Parkinson's Disease Progression Patterns

Careful management of patient's condition is crucial to ensure the patient's independence and quality of life. This is achieved by personalized treatment based on individual patient's symptoms and medical history. The aim of this chapter is to determine patient groups with similar disease progression patterns coupled with patterns of medication changes that lead to the improvement or decline of patients' quality of life symptoms. To this end, this chapter proposes a new methodology for clustering of short time series of patients' symptoms and prescribed medications data, and time sequence data analysis using skip-grams to monitor disease progression. The chapter is divided into two sections. We first introduce the problem description and then we present the published *Journal of Intelligent Information Systems* paper that addresses the described problem.

5.1 Problem Description

Parkinson's disease symptoms affect the quality of life of both the patients and their closest communities. There is no cure for Parkinson's disease and the treatment of patients is directed towards managing their symptoms with antiparkinson medication and prolonging their independence. In time patients begin to lose independence and rely heavily on the support of their families. Both the medical treatment and the patients' inability to play a more active role in their countries economies present a heavy economic pressure on countries worldwide. Early diagnosis and response to patients' symptoms in adequate time are of essential importance to contain the progression and degradation of patients' normal quality of life.

Data mining techniques for Parkinson's disease have been limited to disease diagnosis, symptoms recognition, or subtype detection. An important and still unresearched area is the problem of disease progression. Patients can have different patterns of disease progression depending both on their symptoms status as well as their clinicians' reaction to their status with medications. Determining patterns of disease progression can help clinicians focus their attention on the therapies that have proven to be best for patients covered by a certain disease progression pattern and thus prolong the independence of their patients.

The progression of patients' disease can be followed by monitoring their overall status over time. The PPMI data record patients' symptoms for each visit to the clinician. The combination of symptoms and their severity a patient experiences on a visit gives information about the patient's status at that time. Based on the experienced symptoms at two consecutive visits, patients' status can either improve, degrade, or stay unchanged.

We address the issue of determining the overall status of patients' by dividing them into groups of patients with similar symptoms. We use descriptive methods to determine the severity of patients status in each group. Descriptions of clusters allow for at least partial ordering of patients' status according to the severity of symptoms of their member patients. As the patients' status changes through time, so will the patients' cluster membership. Change of patients' cluster assignment between two consecutive visits indicates a change in their overall status thus drawing the trajectory of their disease progression. Analysis of all trajectories of disease progression will reveal patterns of disease progression shared among Parkinson's disease patients.

As mentioned in Chapter 3, the data describing the status of Parkinson's disease patients is obtained from multiple sources and describes different aspects of the patients' lives. We address the division of patients into groups in two settings where we look into their i) overall status based on the sum of their symptoms severities and ii) the patients' status as presented from multiple angles (multi-view learning). Based on these divisions of patients, we demonstrate that motor and autonomic symptoms are the most informative for evaluating the quality of life of Parkinson's disease patients.

We also address the issue of medications therapy changes. We are interested in patterns of medications changes that influence the improvement or the degradation of the patients' status. By following the evolution of symptoms for each patient separately, we are able to determine patterns of medication changes that can lead to the improvement or degradation of the patients' quality of life.

5.2 Related Publication

The rest of this chapter presents the *Journal of Intelligent Information Systems* paper.

Publication related to this contribution

Journal Paper

Valmarska, A., Miljkovic, D., Lavrač, N., & Robnik-Šikonja, M. (2018). Analysis of medications change in Parkinson's disease progression data. *Journal of Intelligent Information Systems*. doi:10.1007/s10844-018-0502-y[1]

This publication contains the following contributions:

- We present the unaddressed issue of Parkinson's disease progression analysis with data mining.
- We present our clustering-based methodology for determining patterns of disease progression. The methodology can employ both single or multi-view clustering approaches.
- We use external cluster validation measures to determine the best clusters.
- We empirically show that patients can be divided into clusters with similar symptoms. These clusters can be at least partially ordered based on the severity of symptoms of patients assigned to them.

[1]ERRATUM: Figure 4 and Figure 5 on page 75 of the PhD thesis have mistakenly been switched, while the captions of Figure 4 and Figure 5 are correct and correspond to the references in the text.

- We empirically confirm the importance of motor and autonomic symptoms on the patients' overall status.
- We present the reasoning behind following the changes of patients' cluster assignments to determine patterns of disease progression and use skip-grams to obtain robust patterns of disease progression.
- We present patterns of disease progression and the characteristics of patients for a few subjectively chosen patterns of progression.
- We present an algorithm for determining patterns of medications dosage changes that influence the improvement or decline of the patients' status. We present these patterns.

The authors' contributions are as follows. Anita Valmarska initiated the idea of clustering-based analysis of short-time series Parkinson's disease symptoms and medication data to determine patterns of disease progression and therapy modifications. Marko Robnik-Šikonja suggested using skip-grams in order to obtain more robust patterns of disease progression. The methodology was designed and developed by Anita Valmarska with the insights from Marko Robnik-Šikonja. Anita Valmarska implemented the methodology and performed the experimental work. Marko Robnik-Šikonja and Nada Lavrač supervised the implementation of the algorithms. Nada Lavrač has suggested to address the problem of Parkinson's disease data analysis and gave insightful comments on data mining of medical data. Dragana Miljkovic provided insights into data mining of Parkinson's disease data. All authors contributed to the text of the manuscript.

J Intell Inf Syst
https://doi.org/10.1007/s10844-018-0502-y

Analysis of medications change in Parkinson's disease progression data

Anita Valmarska[1,2] · Dragana Miljkovic[1] · Nada Lavrač[1,2,3] · Marko Robnik-Šikonja[4]

Received: 17 July 2017 / Revised: 9 February 2018 / Accepted: 19 February 2018

Abstract Parkinson's disease is a neurodegenerative disorder that affects people worldwide. Careful management of patient's condition is crucial to ensure the patient's independence and quality of life. This is achieved by personalized treatment based on individual patient's symptoms and medical history. The aim of this study is to determine patient groups with similar disease progression patterns coupled with patterns of medications change that lead to the improvement or decline of patients' quality of life symptoms. To this end, this paper proposes a new methodology for clustering of short time series of patients' symptoms and prescribed medications data, and time sequence data analysis using skip-grams to monitor disease progression. The results demonstrate that motor and autonomic symptoms are the most informative for evaluating the quality of life of Parkinson's disease patients. We show that Parkinson's disease patients can be divided into clusters ordered in accordance with the severity of their symptoms. By following the evolution of symptoms for each patient separately, we were able to determine patterns of medications change which can lead to the improvement or worsening of the patients' quality of life.

✉ Anita Valmarska
anita.valmarska@ijs.si

Dragana Miljkovic
dragana.milkovic@ijs.si

Nada Lavrač
nada.lavrac@ijs.si

Marko Robnik-Šikonja
marko.robnik@fri.uni-lj.si

[1] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

[2] Jožef Stefan International Postgraduate School, Jamova 39, 1000 Ljubljana, Slovenia

[3] University of Nova Gorica, Vipavska 13, 5000 Nova Gorica, Slovenia

[4] Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, 1000 Ljubljana, Slovenia

Published online: 20 March 2018

J Intell Inf Syst

Keywords Parkinson's disease · Quality of life indicators · Clustering · Short time series · Skip-grams

1 Introduction

Parkinson's disease is a neurodegenerative disorder that affects people worldwide. Due to the death of nigral neurons, there are changes in dopamine levels in the human brain causing several motor symptoms: tremor, rigidity, bradykinesia and postural instability. In addition to motor symptoms, Parkinson's disease is associated also with non-motor symptoms, which include cognitive, behavioral, and autonomic problems. These symptoms significantly decrease the quality of life of the patients affected by Parkinson's disease.

Over 6.3 million people have the condition worldwide (European Parkinson's Disease Association 2016). In Europe, more than one million people live with Parkinson's disease and this number is expected to double by 2030 (Dorsey et al. 2007). Parkinson's disease is the second most common neurodegenerative disease (after Alzheimer's disease) and its prevalence continues to grow as the population ages. Currently, there is no cure for Parkinson's disease. The reasons for the cell death are still poorly understood. The management of symptoms is of crucial importance for patients' quality of life, mainly addressed with antiparkinson medication, such as levodopa and dopamine agonists.

While numerous studies address specific aspects of the disease, there are few research efforts that adopt a holistic approach to disease management (Gatsios et al. 2016). The PERFORM (Tzallas et al. 2014), REMPARK (Samà et al. 2015) and SENSE-PARK (SENSE-PARK 2016) systems are intelligent closed-loop systems that seamlessly integrate a range of wearable sensors (mainly accelerometers and gyroscopes), constantly monitoring several motor signals of the patients and enabling the prescribing clinicians to remotely assess the status of the patients, given a real-time image of each patient's condition. Based on individual patient's response to the prescribed therapy (manifested by the change of the motor symptoms), the physician is able to adjust medication schedules and personalize the treatment (Gatsios et al. 2016). However, no data mining paradigms are used in the mentioned systems.

In the development of the PD_manager's m-Health platform for patient-centric Parkinson's disease management (PD_manager: m-Health platform for Parkinson's disease management 2015), one of the investigated approaches is data mining, aiming to provide decision support to clinicians and patients in personalized disease management. The individual patient's data, recorded in consecutive visits to the prescribing physician, are collected from different sources offering different 'views' of the data describing the same patient by multiple distinct feature sets. This setting suggests a multi-view learning approach.

Multi-view learning—a relatively new but well-established machine learning technique—is often appropriate for this type of data, as it aims to build models from multiple views (multiple data sets) by considering the diversity of different views (Xu et al. 2013). These views represent data obtained from multiple sources or different feature subsets and describe the same set of examples. We decided for a multi-view clustering approach, aiming to construct disjoint partitioning of objects (patients) described by multiple feature sets. This partitioning is aimed at identifying clusters of patients that share similar symptoms which enables automatic detection of interesting patterns.

Our work explores and tries to give answers to important medical questions which nobody (to the best of our knowledge) has tried to answer: How medications therapy of Parkinson's disease patients changes in response to the patients' change of overall status, and what are the directions in which the disease would develop based on the patients' symptoms and

their therapies. The goal of this paper is to develop a new clustering-based methodology for disease progression data, which will—based on the patients' allocation to clusters at given time points and their history of medication therapies—be able to make suggestions about modifications of particular patient's therapy, with the aim to improve the patient's quality of life. Patients' allocation to clusters represent their disease status, and their changed cluster allocation through time represents their disease progression. The analysis of the clusters can reveal what is the most common status of the patients, and the analysis of cluster changes can reveal how their symptoms change in time. Learning on the history of changes between clusters allows us to infer significant features and relevant medications changes for groups of patients and to suggest medications changes for the individual patients.

In order to increase the robustness of our results, we model the sequences of changes of patient's status between the clusters by using *skip-grams* (Guthrie et al. 2006), an approach upgrading the more standard n-grams approach (Broder et al. 1997) that is regularly used in the analysis of data sequences. The introduction of skip-grams results in increased number of investigated n-grams, providing a more stable distribution of the possible cluster changes.

This paper significantly extends our previous work (Valmarska et al. 2016). We extended the methodology for analysis of Parkinson's disease data to include three threads of clustering (Section 4). A pseudo code of the approach for dividing Parkinson's disease patients into groups with similar symptoms and ordering these groups of patients in accordance with the severity of their overall status is outlined in Section 4.2. The changes of patients antiparkinson medications dosages in relation to the change of their overall status is explored in Section 4.3, where we introduce Algorithm 2 to determine the change in medications dosage with respect to the change of patient's status. In Section 4.4 we present the skip-grams based approach for determining groups of patients with different patterns of disease progression based on the changes of their overall status. Finally, we have significantly extended the empirical evaluation in Section 5 by updating the previous symptoms analysis and medications analysis results with the results for determining the number of clusters and patterns of disease progression. We also present the results of detailed analysis of patients who were identified as following a certain pattern of disease progression.

The paper is structured as follows. After presenting the motivation, the background and the related work in Section 2, Section 3 describes the Parkinson's Progression Markers Initiative (PPMI) data (Marek et al. 2011) used in our experiments. Section 4 proposes the methodology for analyzing the Parkinson's disease data through clustering of short time series symptoms data and connecting the changes of symptoms-based clustering of patients to the changes in medication therapies with the goal to find treatment recommendation patterns and disease progression patterns. The latter is addressed by introducing the so-called skip-grams for analyzing the cluster change patterns and the progression of the disease. Section 5 presents the results of data analysis, tested on two data set variants. Finally, Section 6 presents the conclusions and ideas for further work. The paper contains four appendices which contain detailed results of analyzes: comparison of clustering algorithms (Appendix A), unsupervised feature selection (Appendix B), evaluation of different views in multi-view clustering (Appendix C), and descriptive rules for multi-view clusters (Appendix D).

2 Background

Parkinson's disease is a heterogeneous neurodegenerative condition with different clinical phenotypes, genetics, pathology, brain imaging characteristics and disease duration (Foltynie et al. 2002). This variability indicates the existence of disease subtypes. Moreover,

Parkinson's disease symptoms overlap with symptoms from other diseases, thus hampering the diagnosis of new PD patients and decreasing the overall success of the diagnosis process. Only 75% of clinical diagnoses of Parkinson's disease are confirmed to be idiopathic Parkinson's disease at autopsy (Hughes et al. 1992).

Given the heterogeneous nature of Parkinson's disease (PD), the nature of data describing PD patients is also heterogeneous, possibly gathered in different databases. Our data set (Marek et al. 2011) contains symptoms of patients suffering from Parkinson's disease where the symptoms are divided into several views. We test the union of all views with standard clustering approaches as well as several subsets of views using multi-view clustering in order to identify clusters of patients that share similar symptoms.

Patients' symptoms change through time depending on the received therapies, development of the disease, everyday habits, etc. We treat patients' symptoms at each time point as one training instance. This leads to patients' allocation to different clusters in different time points depending on the progression of the disease. We aim to suggest modifications of the medication treatments based on identified migration patterns of patients from one cluster to another with the goal to keep the patients in the clusters with symptoms that allow a good quality of life. To reach this goal we developed a new clustering-based methodology for disease progression data.

The reminder of this section presents Parkinson's disease related data mining research, an overview of relevant multi-view clustering approaches, and a short overview of methods for short time series analysis, including the introduction of skip-grams for sequence data analysis.

2.1 Parkinson's disease related data mining research

Data mining research in the field of Parkinson's disease (PD) can be divided into four groups: classification of PD patients, detection of PD symptoms, detection of subtypes of PD patients, and assessing success of deep brain stimulation surgery as a last resort in the treatment of Parkinson's disease patients.

The use of classification techniques offers decision support to specialists by increasing the accuracy and reliability of diagnosis and reducing possible errors. Gil and Johnson (2009) use Artificial Neural Networks (ANN) and Support Vector Machines (SVM) to distinguish PD patients from healthy subjects. Ramani and Sivagami (2011) compare the effectiveness of different data mining algorithms in the diagnosis of PD patients.

Tremor is one of the symptoms strongly associated with Parkinson's disease. Several methods for numerical assessment of the intensity of tremor have been proposed. These methods include time series analysis (Timmer et al. 1993), spectral analysis (Riviere et al. 1997) and non-linear analysis (Riviere et al. 1997) and they address tremor detection and quantification. Recent works are based on body fixed sensors (BFS) for long-term monitoring of patients (Patel et al. 2009).

In the course of their disease, patients are prescribed antiparkinson medications therapies in order to control the troubling symptoms. As the disease progresses, the medications treatment can become ineffective and—as a last resort—clinicians use deep brain stimulation (DBS) surgery to control the Parkinson's disease symptoms. Data mining research confirms that DBS significantly improves the patients' motor function (Liu et al. 2014). Depending on the chosen method for DBS, a great reduction in dose of medication, or conservation of cognitive functions can be achieved. In order to predict the neurological effects related to different electrode-contact stimulation, Szymański et al. (2015) tracked the connections between the stimulated part of subthalamic nucleus and the cortex with the help of diffusion tensor imaging (DTI).

Identification of Parkinson's disease subtypes is presented in the work of Lewis et al. (2005), and has been confirmed by the conclusions from Reijnders et al. (2009) and Ma et al. (2015). While clustering usually focuses on patient grouping with the aim of diagnosing new patients, none of the listed methods follows the progression of the disease, and to the best of our knowledge, no data mining research in the field of Parkinson's disease analyzed the development of the disease in combination with the medications that the patients receive. Identification of groups of patients based on the similarity of their symptoms and the clinicians' reaction with medications modification in order to keep the patients as stable and in good status as possible, can be helpful in the assignment of personalized therapies and an adequate patient treatment. For that purpose, we propose a methodology for identification of groups of patients based on the severity of their symptoms, determination of disease progression, and the consequent patterns of medications modifications.

2.2 Multi-view clustering

Multi-view clustering is concerned with clustering of data by considering the information shared by each of the separate views. Many multi-view clustering algorithms initially transform the available views into one common subspace (early integration), where they perform the clustering process (Xu et al. 2013). Chaudhuri et al. (2009) propose a method for multi-view clustering where the translation to a lower vector space is done by using Canonical Correlation Analysis (CCA). Tzortzis and Likas (2009) propose a multi-view convex mixture model that locates clusters' representatives (exemplars) using all views simultaneously. These exemplars are identified by defining a convex mixture model distribution for each view. Cleuziou et al. (2009) present a method where in each view they obtain a specific organization using fuzzy k-means (Bezdek 1981) and introduce a penalty term in order to reduce the disagreement between organizations in the different views. Cai et al. (2013) propose a multi-view k-means clustering algorithm for big data. The algorithm utilizes a common cluster indicator in order to establish common patterns across the views.

Co-training (Blum and Mitchell 1998) is one of the earliest representatives of multi-view learning. This approach considers two views consisted of both labeled and unlabeled data. Using labeled data, co-training constructs a separate classifier for each view. The most confident predictions of each classifier on the unlabeled data are then used to iteratively construct additional labeled training data. Kumar and III (2011) apply the co-training principle (Blum and Mitchell 1998) in unsupervised learning. Clustering is performed on both views, then cluster points from one view are used to modify the clustering structure of the other view. Appice and Malerba (2016) employ the co-training principle in the multi-view setting for process mining clustering. The above-mentioned approaches presume that each of the respective views is capable of producing clusters of similar quality when considered separately. He et al. (2014) do not make that presumption. They combine multiple views under a principled framework, CoNMF (Co-regularized Non-negative Matrix Factorization), which extends NMF (Non-negative matrix factorization) for multi-view clustering by jointly factorizing the multiple matrices through co-regularization. The matrix factorization process is constrained by maximizing the correlation between pairs of views, thus utilizing information from each of the considered views. CoNMF is a multi-view clustering approach with intermediate integration of views, where different views are fused during the clustering process. The co-regularization of each pair of views makes the clustering process more robust to noisy views. The decision to use the CoNMF approach in our work was made based on this algorithm property and on the availability of its Python code.

2.3 Analysis of short time series

A time series is a series of data points indexed in time order. Time series data analysis was used to study a wide range of biological and ecological systems (Bence 1995). The use of time series allows for studying the dynamics of a system. Short time series (8 points or less) constitute more than 80% of all time series data sets (Ernst et al. 2005). The small number of available time points does not allow for identification of statistically significant temporal profiles (Ernst and Bar-Joseph 2006). Bence (1995) examines methods for adjusting confidence intervals of the mean and parameters of a linear regression for autocorrelation. De Alba et al. (2007) suggest that simpler models can be more effective on short time series. They show that the Bayesian approach is superior to the traditional approach when applied to short time series but inferior when applied on longer time series (De Alba et al. 2007). Most of the research in short time series analysis is related to the analysis of short time series microarray gene expression data. Ernst et al. (2005) present a method for clustering of short time series gene expression data, followed by the introduction of the STEM (Short Time-series Expression Miner) software program (Ernst and Bar-Joseph 2006) specifically designed for the analysis of short time series microarray gene expression data.

In the healthcare domain, Choi et al. (2017) incorporate temporal modeling using the recurrent neural network (RNN) model to predict heart failure. Imhoff et al. (1998) apply short time series analysis to monitor lab variables after liver surgery, and to offer support to clinicians in their decision-making process for the treatment of acute respiratory distress syndrome. Schieb et al. (2013) evaluate the clustering of stroke hospitalization rates, patterns of the clustering over time, and associations with community level characteristics. They generate clusters of high and low-stroke hospitalization rates during two periods of time. According to the place of residence of patients, counties in USA are assigned to a cluster. Following the transition of counties between clusters between these two periods, counties are labeled as having a persistently high, transitional, or persistently low-stroke hospitalization rate.

Murugesan et al. (2017) present a hierarchical multi-scale approach for visualizing spatial and functional cluster evaluation patterns. Their visualization method is two-stage method based on sequence of community detection at each time stamp and community tracking between steps. Greene et al. (2010) address the issue of identifying communities in dynamic networks. Appice (2017) uses social network analysis as a basic approach for organizational mining, aimed at understanding the life cycle of a dynamic organizational structures.

Zhao et al. (2017) explore different representations of temporal data from electronic health records to improve prediction of adverse drug events. They obtain sequences of symbols by transforming time series of individual feature into strings (Lin et al. 2007). These strings reflect the temporal nature of the original values. Results from their empirical investigation show that transformation of sequences to tabular form based on edit distance of sub-sequences to representative shaplets leads to improvements in the predictive performance. This approach reduces the feature sequence diversity by finding informative random sub-sequences. The goal of Zhao et al. (2017) is to predict whether patients will develop adverse drug reactions. They use the history of patients symptoms in order to predict a single event (adverse drug event: yes or no), while we follow the patients' disease development and changes in their overall status as a result of therapy changes. Another difference is our use of skip-grams which reduces noise and enforces strong transition patterns.

To the best of our knowledge, the temporal nature of medical data has not been explored in research directed toward determining the progression of a particular disease and determining the therapy recommendations in order to stabilize the disease progression. We present a clustering based methodology on short time series symptoms data of Parkinson's disease

patients in an attempt to discover how the disease develops through time, reflected by the change of patients' symptoms. Simultaneously, we use the temporal data about their medications therapy to determine how clinicians react to patients' symptoms changes. Each Parkinson's disease patient is described with his/her symptoms and medications treatment through time. The temporal data is flattened to records from single time points, referred in this manuscript as instances, where any change of patients' symptoms between two consecutive points is referred as change in their status. Changes in status are then connected to possible changes in medications therapies.

2.4 Skip-grams for sequence data analysis

Patient's allocation to clusters in sequential time points can be viewed as a sequence of items. Analysis of contiguous sequences of items for every patient's cluster allocation can provide an insight into the disease progression and reveal patterns how (and how often) the patient's symptoms improve or degrade.

In this paper we use an approach to sequence data analysis, where we borrow the methodology initially developed in the field of natural language processing (NLP). In NLP, a contiguous sequence of n items from a given sequence of text or speech is called an n-gram (Broder et al. 1997). Skip-grams are a generalization of n-grams in which the components (typically words) need not be consecutive in the text under consideration but may leave gaps that are skipped over (Guthrie et al. 2006). They provide a way of overcoming the data sparsity problem found with conventional n-gram analysis.

Another use of skip-grams is in producing word embeddings into a vector form to reduce dimensionality and sparsity of bag-of-words representation. Mikolov et al. (2013) proposed word2vec embedding based on deep learning, which has subsequently been used in many NLP applications, including some with clinical text data (Minarro-Giménez et al. 2013; De Vine et al. 2014) (PubMed abstracts, disease progression reports) and to learn relationships between clinical processes or unified medical language system (UMLS) concepts (Choi et al. 2017). Our use of skip-grams is entirely different as we do not use embeddings but use skip-grams directly as a more robust version of n-grams.

In the context of our analysis, skip-grams allow for robust identification of frequent paths through clusters and reveal typical disease progression patterns. The patient's overall status at a given visit to the clinician, as determined by the (patient, visit) pair cluster assignment, can be seen as an item, and changes of clusters as sequences of items, which can be analyzed with the skip-grams based approach developed in NLP. This is novel in the analysis of Parkinson's disease data and allows us to follow the progression of the patient's overall status without taking into account noise in the form of sudden changes in the patient's status. Such changes are not necessary due to Parkinson's disease, but can be attributed to other stressful events in the patient's life (such as loss of a pet, loss of a loved one, etc). To the best of our knowledge, there has not been any study involving skip-grams that uses the actual symptoms of patients in order to explore patient's disease progression and the clinicians' response by changing the medications therapy. A formal definition of skip-grams and their use are presented in Section 4.4.

3 Data

In this study, we use the PPMI data collection (Marek et al. 2011) gathered in the observational clinical study to verify progression markers in Parkinson's disease. The PPMI data

collection consists of data sets describing different aspects of the patients' daily life. Below we describe the selection of PPMI data used in the experiments.

3.1 PPMI symptoms data sets

The medical condition and the quality of life of a patient suffering from Parkinson's disease can be determined using the Movement Disorder Society (MDS) sponsored revision of Unified Parkinson's Disease Rating Scale (MDS-UPDRS) (Goetz et al. 2008). It is a questionnaire consisting of 65 questions concerning the progression of disease symptoms. MDS-UPDRS is divided into four parts. Part I consists of questions about the 'non-motor experiences of daily living'. These questions address complex behaviors, such as hallucinations, depression, apathy, etc., and patient's experiences of daily living, such as sleeping problems, daytime sleepiness, urinary problems, etc. Part II expresses 'motor experiences of daily living'. This part of the questionnaire examines whether the patient experiences speech problems, the need for assistance with the daily routines such as eating or dressing, etc. Part III is referred to as the 'motor examination', while Part IV concerns 'motor complications', which are mostly developed when the main antiparkinson drug levodopa is used for a longer time period. Each question is anchored with five responses that are linked to commonly accepted clinical terms: 0 = normal (patient's condition is normal, symptom is not present), 1 = slight (symptom is present and has a slight influence on the patient's quality of life), 2 = mild, 3 = moderate, and 4 = severe (symptom is present and severely affects the normal and independent functioning of the patient, i.e. her quality of life is significantly decreased).

Montreal Cognitive Assessment (MoCA) (Dalrymple-Alford et al. 2010) is a rapid screening instrument for mild cognitive dysfunction. It is a 30 point questionnaire consisting of 11 questions, designed to assess different cognitive domains: attention and concentration, executive functions, memory, language, visuoconstructional skills, conceptual thinking, calculations, and orientation.

Scales for Outcomes in Parkinson's disease – Autonomic (SCOPA-AUT) is a specific scale to assess autonomic dysfunction in Parkinson's disease patients (Visser et al. 2004). Physical Activity Scale for the Elderly (PASE) (Washburn et al. 1993) is a questionnaire which is a practical and widely used approach for physical activity assessment in epidemiologic investigations. The above data sets are periodically updated to allow the clinicians to monitor patients' disease development through time. Answers to the questions from each questionnaire form the vectors of attribute values.

Table 1 summarizes the symptoms data sets considered in our research. It lists the number of considered questions from each questionnaire, the range of attribute values, and the nature of the attribute values. All of the considered questions have ordered values, and—with the exception of questions from MoCA and PASE—increased values suggest higher symptom severity and decreased quality of life.

When considering the possibility of using a multi-view framework, the independence of the separate views should be inspected. In their work, Goetz et al. (2008) present that the MDS-UPDRS shows high internal consistency (Cronbach's alpha = 0.79—0.93 across parts). MDS-UPDRS across-part correlations range from 0.22 to 0.66. Reliable factor structures for each part are obtained (comparative fit index > 0.90 for each part), which supports the use of sum scores for each part, when compared to using a total score of all parts.

3.2 PPMI concomitant medications log

The PPMI data collection offers information about all of the concomitant medications that the patients used during their involvement in the study. These medications are described

Table 1 Characteristics of the questionnaire data used in the analysis

Questionnaire	Number of questions	Answers value range	Ordered values	Higher value indicates higher symptom severity
MDS-UPDRS Part I	6	0–4	Yes	Yes
MDS-UPDRS Part Ip	7	0–4	Yes	Yes
MDS-UPDRS Part II	13	0–4	Yes	Yes
MDS-UPDRS Part III	35	0–4	Yes	Yes
MoCA	11	0–1	Yes	No
PASE	7	1–2	Yes	No
SCOPA-AUT	21	0–3	Yes	Yes

by their name, the medical condition they are prescribed for, as well as the time when the patient started and (if) ended the medications therapy. For the purpose of our research, we initially concentrate only on whether the patient receives a therapy with antiparkinson medications, and which combination of antiparkinson medications the patient has received between each of the time points when the MDS-UPDRS test and the MoCA test were administered. The main families of drugs used for treating motor symptoms are levodopa, dopamine agonists and MAO-B inhibitors (National Collaborating Centre for Chronic Conditions 2006). Medications which treat Parkinson's disease-related symptoms but are not from the above-mentioned groups of medications are referred to as *other*.

3.3 Experimental data

Symptoms of patients suffering from Parkinson's disease are grouped into several data sets, representing distinct views of the data. These views consist of data from MoCA test, motor experiences of daily living, non-motor experiences of daily living, complex motor examination data, etc. For each patient these data are obtained and updated periodically (on each patient's visit to the clinician)—at the beginning of the patient's involvement in the PPMI study, and approximately every 6 months, in total duration of 5 years—providing the clinicians with the opportunity to follow the development of the disease. The visits of each patient can be viewed as time points, and the collected data on each visit is the data about the patient in the respective time point. All time points collected for one patient form a short time series.

In the experiments we address two settings: the analysis of *merged symptoms data* and the analysis of *multi-view symptoms data*.

Merged symptoms data are represented in a single data table, constructed by using the sums of values of attributes of the following data sets: MDS-UPDRS Part I (subpart 1 and subpart 2), Part II, Part III, MoCA, PASE, and SCOPA-AUT.[1] Goetz et al. (2015) use sums of symptoms values as an overall severity measure of a given aspect of Parkinson's disease. Similarly, we use sums of attribute values from different data sets to present the overall status of patients concerning respective aspects of their everyday living. Table 2 outlines the attributes used to construct the merged symptoms data, together with their range of values. This is a simplified representation using seven attributes, each representing the severity of symptoms of a given symptoms group, which proved to be valuable in the initial experiments (Valmarska et al. 2016).

[1] Appendix B presents the clustering quality results on data set obtained by feature selection.

Table 2 List of attributes used in the merged symptoms data set

Dataset	Attribute name	Value range	Higher value indicates higher symptom severity
MDS-UPDRS Part I	NP1SUM	0–24	Yes
MDS-UPDRS Part Ip	NP1PSUM	0–28	Yes
MDS-UPDRS Part II	NP2SUM	0–52	Yes
MDS-UPDRS Part III	NP3SUM	0–138	Yes
MoCA	MCATOT	0–30	No
PASE	PASESUM	0–24	No
SCOPA-AUT	SCAUSUM	0–63	Yes

Multi-view symptoms data consist of seven data sets: MDS-UPDRS Part I, Part Ip, Part II, Part III, MoCA, SCOPA-AUT, and PASE. Each of these data sets consists of values of attributes, which represent answers to the questions from a particular questionnaire. Similarly to Goetz et al. (2015), we added an additional attribute to each data set, which is the sum of values of attributes in the given data set (this equals the values of individual attributes used in the merged symptoms data).

The experimental data include symptoms and medications data of 405 Parkinson's disease patients from the PPMI study. Out of these 405 patients, 265 patients are male and 140 are female. The youngest patient was 33 years old at the beginning of the study (baseline visit), and the oldest patient was 84 years old. The average age of patients is 61.09 years. The experimental data contains from 1 to 5 visits to the clinician. The average number of recorded visits is 3.321. The experimental data consist of 1,345 patient's visits and each visit is considered a separate data instance, representing the basic building block of the methodology described in Section 4.

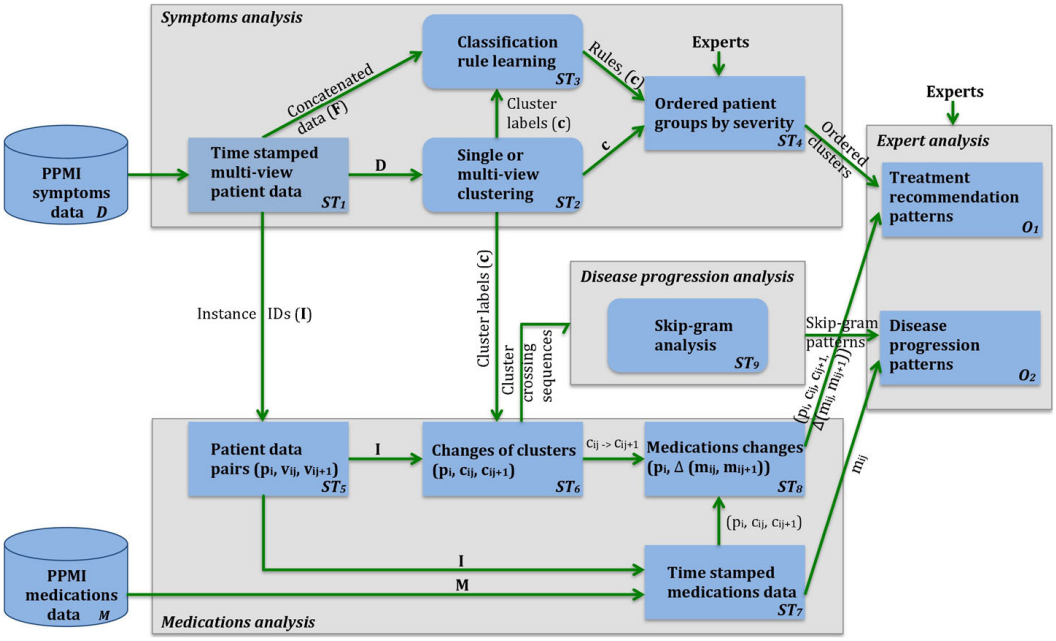


Fig. 1 Outline of the approach to Parkinson's disease quality of life data analysis

4 Methodology

To assist the clinicians in making decisions regarding the patients' therapy, we propose a procedure which involves a combination of clustering patients' symptoms data and the analysis of histories of patients' medication treatments, followed by disease progression analysis. Figure 1 shows an outline of the proposed methodology, which addresses changes of data over time (i.e. over several patient's visits) with the goal to suggest possible modifications of the medication treatment. Moreover, our goal is to analyze the rate of progression of Parkinson's disease and discover the most frequent patterns of symptoms change; we address this goal by using skip-grams on patients' changes of clusters. The usage of skip-grams can reveal groups of patients with an unusual pattern of symptoms change which deserve a more thorough look into the characteristics of that groups.

The input to the methodology are PPMI data sets of patient symptoms (described in Section 3.1) and the PPMI medications log data (described in Section 3.2), and the outputs are treatment recommendation patterns that can assist the clinician in deciding about further treatment of a patient, as well as the disease progression patterns providing insight into disease development. The methodology[2] consists of three separate threads whose outputs are combined to identify treatment recommendation patterns and disease progression patterns.

Symptoms analysis. The first thread, referred to as *Symptoms analysis* in the top part of Fig. 1, finds groups of patients with similar symptoms by grouping the instances, defined as (patient, visit) pairs. It uses clustering and describes the discovered patient groups with induced classification rules where classes correspond to individual cluster labels. Details of this thread are presented in Section 4.2.

Medications analysis. The second thread, referred to as *Medication analysis* in the bottom part of Fig. 1, is concerned with finding changes of medications and their dosages based on patients' symptoms changes between two consecutive visits to the clinician (e.g., disease aggravation, improvement or no change). In this thread we observe the patients moving from one cluster to another cluster in two consecutive time points, i.e. two consecutive visits to the clinician. The outcomes of the two threads are combined to a set of treatment recommendation patterns (i.e. increased/decreased/unchanged dosage of medications) for the four groups of medications mentioned in Section 3.2. We elaborate on this thread in Section 4.3.

Disease progression analysis. The third thread, referred to as *Disease progression analysis* in the middle part of Fig. 1, is concerned with finding patterns of disease progression, using skip-grams analysis on cluster crossing sequences. Details are given in Section 4.4.

The first step of the methodology is the construction of individual patient-visit pairs (p_i, v_{ij}), representing individual instances or items. For each patient p_i a set of pairs (p_i, v_{ij}) is constructed, where v_{ij} describes the symptoms recorded at an individual patient's visit to the clinician. These instances (patient-visit pairs) are the items representing the basic unit of analysis in the *Symptoms analysis* thread of the methodology. The attribute values of instance (p_i, v_{ij}) correspond to symptoms of patient p_i on visit j, and v_{ij} and v_{ij+1} correspond to two consecutive patient's visits. This is followed by clustering of instances.

The basic unit of the *Medications analysis* thread of the methodology are $(p_i, v_{ij}, c_{ij}, m_{ij}, v_{ij+1}, c_{ij+1}, m_{ij+1})$ tuples, where c_{ij} is the cluster label for instance (p_i, v_{ij}) and m_{ij} are the medications that patient p_i takes at the time of visit j. Elements c_{ij+1} and m_{ij+1} are

[2]The code is available upon request. Please note, we do not have a permission to share the data. Users can obtain permission from the Parkinson's Progression Markers Initiative (PPMI): http://www.ppmi-info.org/

the cluster label and prescribed medications of the same patient on visit $j + 1$, i.e. at the time of the next visit.

The basic unit of the *Disease progression analysis* thread are patients' sequences of cluster crossings. Patient p_i cluster crossing sequence is Seq_i, defined as a sequence of cluster assignments for patient p_i at time points $v_{i1}, v_{i2}, ..., v_{ik_i}$, where v_{ij} correspond to the symptoms recorded at visit v_{ij} of patient p_i, and k_i is the number of visits to the clinician by patient p_i.

As our methodology is based on clustering, in Section 4.1 we first present cluster validity indices used to asses the quality of clusters produced by different tested methods.

4.1 Cluster validity indices

The number of groups (clusters) of similar patients was unknown before the start of the data analysis. In order to estimate the optimal number of clusters, we used internal cluster validity indices (Arbelaitz et al. 2013), which are—in the absence of ground truth labels—used to estimate the quality of generated clusters. The clustering quality is determined based on cluster *compactness*—how close are the related objects in each cluster, and cluster *separation*—how distinct or well-separated is each cluster from other clusters.

Many clustering validity indices (i.e. cluster quality measures) exist. We use three of the best performing indices from Arbelaitz et al. (2013): Silhouette analysis index (SA) (Rousseeuw 1987), Davies-Bouldin index (DB) (Davies and Bouldin 1979), and Calinski-Harabasz index (CH) (Caliński and Harabasz 1974). Below we present definitions and intuition behind these indices.

Let data set X be a set of N objects represented as vectors in an F-dimensional space, $X = \{x_1, x_2, ..., x_N\} \subseteq \Re^F$. Clustering of X is a set of disjoint clusters that partitions X into K groups. Clustering C is defined as disjoint partition of objects in X, $C = \{c_1, c_2, ..., c_K\}$, where $\bigcup_{c_k \in C} c_k = X$, $c_k \cap c_l = \emptyset$, $\forall k \neq l$. Centroid of a cluster c_k is defined as $\overline{c_k} = \frac{1}{|c_k|} \sum_{x_i \in c_k} x_i$. Similarly, the global centroid is defined as $\overline{X} = \frac{1}{N} \sum_{x_i \in X} x_i$. The Euclidean distance between two objects x_i and x_j is denoted as $d_e(x_i, x_j)$ (Arbelaitz et al. 2013).

Silhouette index is a normalized summation-type index. The compactness is measured based on the distance between all the objects in the same cluster and the separation is based on the nearest neighbor distance (Arbelaitz et al. 2013; Rousseeuw 1987; Kaufman and Rousseeuw 1990).

$$SA(C) = \frac{1}{N} \sum_{c_k \in C} \sum_{x_i \in c_k} \frac{b(x_i, c_k) - a(x_i, c_k)}{\max\{a(x_i, c_k), b(x_i, c_k)\}} \tag{1}$$

where

$$a(x_i, c_k) = \frac{1}{|c_k|} \sum_{x_j \in c_k} d_e(x_i, x_j) \tag{2}$$

is the normalized distance of object x_i to all the objects in the same cluster (low values of this term are indicators of high compactness), and

$$b(x_i, c_k) = \min_{c_l \in C \setminus c_k} \left\{ \frac{1}{|c_l|} \sum_{x_j \in c_l} d_e(x_i, x_j) \right\} \tag{3}$$

is the normalized distance from object x_i to all objecs from its closest neighbor cluster (high values of this term are indicators of high separation). For each object, the quotient in (1) is a value between -1 and 1. A value close to 1 indicates that the object is well placed in its current cluster, while a value close to -1 indicates that it would be better placed in

the nearest cluster. Value 0 indicates a borderline quality of placement. An average over all objects gives an estimate on the overall quality of clusters. If there are too many or too few clusters as a result of inappropriate choice of the number of clusters K, many quotients will be low and the average score will reflect that.

Davies-Bouldin index estimates the compactness based on the distance of objects in a cluster to its centroid and the separation based on the distance between centroids (Arbelaitz et al. 2013; Davies and Bouldin 1979).

$$DB(C) = \frac{1}{K} \sum_{c_k \in C} \max_{c_l \in C \setminus c_k} \left\{ \frac{S(c_k) + S(c_l)}{d_e(\overline{c_k}, \overline{c_l})} \right\} \tag{4}$$

where

$$S(c_k) = \frac{1}{|c_k|} \sum\nolimits_{x_i \in c_k} d_e(x_i, \overline{c_k}) \tag{5}$$

is an average distance from objects in a cluster to its centroid. The $S(c_k)$ is a measure of the compactness for cluster c_k (the lower the value the more compact is the cluster). The quotients in (4) are indicators of separations between two clusters, c_k and c_l (the lower the quotient the better the two clusters are separated). By taking the maximum over these quotients we get the estimation of the worst case separation (i.e. for cluster c_k and its closest cluster). The average over these maxima is the value of DB index, whose lower values indicate better clusterings.

Calinski-Harabasz (CH) index estimates the compactness based on the distances from the objects in a cluster to its centroid (see the denominator below). The separation is based on the distance from the centroids to the global centroid $\overline{X}$ (see the nominator) (Arbelaitz et al. 2013; Caliński and Harabasz 1974).

$$CH(C) = \frac{N - K}{K - 1} \frac{\sum_{c_k \in C} d_e(\overline{c_k}, \overline{X})}{\sum_{c_k \in C} S(c_k)} \tag{6}$$

Factor $K - 1$ normalizes the distances of cluster centroids to global centroid, and factor $N - K = \sum_{k=1}^{K} (|c_k| - 1)$ normalizes the distances of objects to their centroids. Good clustering should have a large value in the nominator (large distances of clusters to global centroid) and a low value in the denominator (low distances of objects to their centroids) and therefore a large value of the CH score.

4.2 Symptoms analysis methodology

After constructing the instances—i.e. (patient, visit) pairs—in step $\mathbf{ST}_1$ of the methodology, the symptoms analysis thread (top of Fig. 1) consists of three further steps: clustering, rule learning and cluster ordering, corresponding to the individual steps of Algorithm 1 (lines 1-3).

The main input to Algorithm 1 is a set of symptoms views $\mathbf{D}$, describing the same n instances, which hold the symptoms data about p patients. This collection consists of m data sets (views). The k-th view ($1 \leq k \leq m$) is defined as $\mathbf{D}_k$, which is a matrix with n rows (the number of instances) and $|A_k|$ attributes. The concatenated data set, denoted as $\mathbf{F}$, is a matrix consisting of n rows and $\sum |A_k|$ columns (union of attributes across all the views). The medication data set, denoted by $\mathbf{M}$, is a matrix consisting of n rows and 4 columns—i.e. dosage data about the 4 PD medication groups. An auxiliary input is $\mathbf{I}$, a matrix which holds the indices of instances, defining the (p_i, v_{ij}) pairs.

The outputs of the algorithm are the assigned cluster labels $\mathbf{c}$ (vector of length n). The clustering of patients uses the provided views (symptoms data sets) and the chosen

clustering method (line 1 and line 2). The probability scores of dosage change of Parkinson's disease medications when patients' statuses improve or degrade are computed in line 4 with Algorithm 2) and are part of the medications change analysis methodology. These probability scores are estimates of medication impact on the change of symptoms.

Algorithm 1 PD_manager medications change methodology

Input : $\mathbf{D}=\{\mathbf{D}_1, ..., \mathbf{D}_m\}$ – collection of views, describing the instances;
m – number of data sets (views);
n – number of instances;
p – number of patients;
$|A_k|$ – number of attributes in k-th data set, $1 \leq k \leq m$;
$\mathbf{D}_k$ – a single view: matrix with n rows and $|A_k|$ columns;
$\mathbf{F}$ – concatenated data: matrix with n rows and $\sum |A_k|$ columns;
$\mathbf{A}=\{A_1 ... A_m\}$ – attribute space;
$\mathbf{M}$ – medications data set with 4 PD medications groups:
matrix with n rows and 4 columns;
$\mathbf{I}$ – indices of patient-visit pairs;

Parameters: clusMethod – clustering method (*default*: if m=1 then k-means
else CoNMF(pairwise));
descMethod – cluster description method (*default*: classification
rule learning);

Output : $\mathbf{c}$ – assigned cluster labels (vector of size n);

```
// Form groups of similar instances.
1 if m=1 then c ← performSingleViewClustering(D,clusMethod);
2 else c ← performMultiViewClustering(D,clusMethod);

// Describe clusters and rank them according to quality of life
   indicators. Ranking is performed by the expert.
3 oClus ← describeClusters(F,c,descMethod)

// Obtain impact of medications.
4 medsChangeProb, medsChange ← getMedsChangeProbabilities(p,M,I,c)

// Obtain medications change patterns for cluster changes classified
   as positive or negative separately.
5 medsModificationPatterns ← getSummedMedsPatterns(medsChange,oClus)
6 visualize(medsModificationPatterns)
```

In step $\mathbf{ST}_2$ of the methodology outlined in Fig. 1, we perform clustering on instances i.e. patient's i symptoms recorded at a visit $v_{,ij}$ in order to determine groups of patients with similar symptoms. Note that the clustering step is performed once on the collection of views $\mathbf{D}$ which describes the instances. Our methodology can address both the merged symptoms data and the multi-view data analysis setting. The only difference is the clustering method applied in step $\mathbf{ST}_2$ of the methodology. In the case of merged symptoms data we performed k-means clustering (line 1 in Algorithm 1), while for clustering of the multi-view data we used the multi-view clustering approach proposed in He et al. (2014) (line 2 in Algorithm 1).

In the next step, $\mathbf{ST}_3$, we use the cluster labels ($\mathbf{c}$) as classes in rule learning in order to obtain meaningful descriptions of patients in each cluster (step $\mathbf{ST}_3$, line 3 in Algorithm 1).

Cluster labels obtained in step $\mathbf{ST}_2$ are input to step $\mathbf{ST}_3$ and are used as class labels in the rule learning process. The purpose of rule learning in step $\mathbf{ST}_3$ is to induce explanatory rules describing the induced clusters. These rules are presented to the experts (step $\mathbf{ST}_4$) to evaluate whether the induced clusters make sense and to determine an ordering of clusters according to the severity of symptoms of instances assigned to them. The rule sets describing the data are induced on a concatenated data set consisting of data sets considered in the clustering step ST_2 (input **F** of Algorithm 1).

The rule sets for each class variable are learned using our recently developed DoubleBeam-RL algorithm (Valmarska et al. 2017; Valmarska et al. 2015). This is a separate-and-conquer classification rule learning algorithm which uses two beams and separate heuristics for rule refinement and rule selection. Stecher et al. (2014) showed that the two phases of rule learning, rule refinement and rule selection, should be separated and use different rule evaluation heuristics in order to obtain rules with improved quality. They also introduce the idea of using the so-called *inverted heuristics* in the refinement phase in order to obtain rules that maximize the number of covered positive examples. By using the heuristics that take full advantage of the refinement and selection process separately, the DoubleBeam-RL algorithm is able to find rules which maximize the number of covered positive examples and minimize the number of covered negative examples, which is the goal of classification rule learning algorithms (Stecher et al. 2014). The DoubleBeam-RL algorithm generates rules with comparable accuracy to the rules generated by the state-of-the-art algorithms for classification rule learning (Valmarska et al. 2017), but as a side effect of using the inverted heuristics in the refinement phase, the induced rules have more conditions. The resulting longer rules with improved expressive power (Stecher et al. 2014; Michalski 1983) are preferred by the clinicians (Gamberger and Lavrač 2002). This is the reason for choosing the DoubleBeam-RL algorithm as the description tool in step $\boldsymbol{ST}_3$. Note that the DoubleBeam-RL algorithm does not perform rule pruning.

In the final step, $\mathbf{ST}_4$, the experts are presented with the descriptions of the obtained clusters, where the expert knowledge is used to interpret the obtained groups of patients and to order them according to the severity of symptoms exhibited by the patients assigned to them. The produced ordering of clusters may be total (all pairs of clusters are comparable) or partial (some clusters may not be comparable). Our methodology works for both cases as described below, but if in this step we get many incomparable clusters, this may be an indication that we have too many irrelevant or redundant attributes and we shall employ feature subset selection.

Based on the expert's interpretation of clusters and the ordering it produces, we take into account only comparable clusters and consider these cluster changes to be either positive or negative. When a patient moves from a cluster described by symptoms indicating worse quality of life to the one described by better quality of life indicators, we consider this change to be positive. A negative cluster change occurs when the symptoms of a patient degrade. Transitions between incomparable clusters are left out of our analysis.

In $\boldsymbol{O}_1$, we combine detected medications changes from step $\boldsymbol{ST}_8$ and cluster severity information from step $\mathbf{ST}_4$. The combined information contains medications changes for positive cluster changes and for negative cluster changes i.e. medications changes with improvement or aggravation of the patients' symptoms. Cluster changes are determined in line 5 of Algorithm 1 and the approach is further explained in Section 4.3.

4.3 Medications change analysis methodology

In this thread of the methodology (bottom of Fig. 1, lines 4–6 in Algorithm 1) we determine the medications changes that have occurred simultaneously with moves between clusters

observed in patients during two consecutive time points (two consecutive visits). An important benefit of our approach is that each patient provides a context (similar observed and unobserved variables) for himself/herself. By following the development of symptoms for each patient separately, we remove the influence of other conditions the patient is treated for.

The information about patients' assignment to clusters and their medication therapy in two consecutive visits is held in $(p_i, v_{ij}, c_{ij}, m_{ij}, v_{ij+1}, c_{ij+1}, m_{ij+1})$ tuples. In step $\boldsymbol{ST}_8$ on Fig. 1 we follow all patients through time. For each pair of patient's p_i consecutive visits to the clinician, we record the cluster change that has occurred between the two visits, $c_{ij} \rightarrow c_{ij+1}$, as well as the change in medications prescriptions, $\Delta(m_{ij}, m_{ij+1})$, which the patients received in the consecutive time points. For each antiparkinson drug group (levodopa, dopamine agonists, MAO-B inhibitors, and others) we record whether their dosage has increased, decreased or stayed unchanged between the two visits. Dosages of PD medications are translated into a common Levodopa Equivalent Daily Dosage (LEDD) which allows for comparison of different therapies (different medications with personalized daily plans of intake).

Algorithm 2 presents the pseudocode of the $getMedsChangeProbabilities$ function. It describes how we determine the changes of medications dosages co-occurring with shifts in patients' symptoms (characterized by a change of clusters). This function (called in line 4 of Algorithm 1) estimates the probability score of medications dosage changes when patients' symptoms have changed (patients have crossed clusters) or stayed the same (patient did not change clusters between two consecutive visits). Additionally, it also counts the type of medication modifications for each cluster crossing. The algorithm takes as an input patients' medications data **M**, the index data set **I**, and the assigned cluster labels **c**. The output are two matrices, $medsChangeProb$ and $medChange$ of the dimension $K \times K \times 4 \times 3$ (K is the number of clusters, we have 4 medication groups and 3 possible changes in severity of symptoms). Each cell of the output matrix $medsChangeProb$ contains a probability that a given medication group will change value (increase, decrease, or stay unchanged) for a certain cluster crossing. Similarly, the $medsChange$ matrix contains the number of changes of each type for each group and each crossing.

For each patient (line 5 in Algorithm 2), we track his/her status development through time. For each two consecutive visits (line 7), we register the clusters the patients were assigned to (lines 8 and 9). These consecutive cluster assignments represent a so-called *cluster crossing* (line 10). For each patient, we also follow therapy changes between two consecutive visits (lines 11 and 12). We consider therapy changes to be dosage changes of any of the antiparkinsonian medications (line 13). For each medications group, we record whether the LED dosage between two consecutive time has *increased*, *decreased*, or stayed *unchanged* (line 14). We record the number of therapy changes for each cluster crossing (line 15). The probability of medications change is calculated in line 24 of Algorithm 2 as the ratio between the recorded number of therapy modifications per cluster crossing and the number of cluster crossings. The output of Algorithm 2 are two matrices, $medsChangeProb$ and $medsChange$, described above.

Both matrices are returned to Algorithm 1. Matrix $medsChange$ is further processed in line 5 with function $getSummedMedsPatterns$. Based on the clusters ordered by the experts according to the severity of symptoms and the information on medications changes for each cluster crossing, we determine patterns of medications adaptations, related to the improvement or aggravation of patients' symptoms. Cluster crossings are classified as either positive or negative. We aggregate (sum) the medications change patterns from cluster changes of the same nature (positive or negative) to determine the patterns of medication modifications when the patients' status improved or worsened. The results are visualized in line 6 of Algorithm 1 (for the results, see Fig. 3).

Algorithm 2: The procedure estimates the probability of medications changes due to symptom changes

```
getMedsChangeProbabilities(p,M, I,c):
  Input      : p – number of patients;
               M – patients' medications data;
               I – indices of patient-visit combinations;
               c – assigned cluster labels;
  Parameters : K – number of clusters in c;
  Output     : medsChangeProb;
               medsChange;
  // Initialize number of cluster changes to 0. Number of clusters is K.
  // noOfCrossings; matrix with K rows and K columns
  noOfCrossings [1:K, 1:K] ← 0
  // Initialize number of medications changes for each cluster change to 0.
  // medsChangeNo; array of dimension K × K × 4 × 3 (4 medication groups, 3 changes).
  medsChangeNo [1:K, 1:K, 1:4, 1:3] ← 0
  // Initialize probabilities of medications changes for each cluster change to 0.
  // medsChangeProb; array of dimension K × K × 4 × 3 (4 medication groups, 3 changes).
  medsChangeProb [1:K, 1:K, 1:4, 1:3] ← 0
  // For each patient check how cluster labels changed between consecutive time points.
  for patient p_i in [1 : p] do
      // consecutive visits for a given patient
      patientsVisits ← I[p_i]
      for v_j, v_{j+1} in patientsVisits do
          // Patient's cluster assignement in two consecutive visits.
          prevCluster ← c[p_i][v_j]
          currCluster ← c[p_i][v_{j+1}]
          // Increase the number of crossings between prevCluster and currCluster.
          noOfCrossings[prevCluster,currCluster]+=1
          // Patient's medications therapy in two consecutive visits (time points).
          prevMeds ← M[p_i][v_j]
          currMeds ← M[p_i][v_{j+1}]
          // Compare medications therapy change between two consecutive visits.
             Patients can receive medications from four medications groups: levodopa,
             dopamine agonist, MAO-B, or other.
          for medsGroup in [1:4] do
              // dosage change can be either increased, decreased or unchanged
              change ← getChange(medsGroup, prevMeds, currMeds) medsChangeNo[prevCluster,
              currCluster,medsGroup, change]+=1
          end
      end
  end
  // Get medications change probability for each of increased, decreased or unchanged.
  // Inspect all cluster crossings and medication groups.
  for c_1 in [1:K] do
      for c_2 in [1:K] do
          clusterCrossings = noOfCrossings[c_1, c_2]
          for medsGroup in [1:4] do
              for change in [1:3] do
                  medsChange = medsChangeNo[c_1, c_2,medsGroup,change]
                  medsChangeProb[c1, c2,medsGroup, change] = medsChange / clusterCrossings
              end
          end
      end
  end
  return medsChangeProb, medsChange
```

4.4 Disease progression analysis using skip-grams

In this thread of the methodology (middle of Fig. 1, step $\boldsymbol{ST}_9$) we determine patterns of cluster changes, resulting in $\boldsymbol{O}_2$ combining the patients' cluster change patterns with the patients' medication data. This allows us to determine patterns of disease progression (indicated by the patterns of cluster change) and impact of medications on these patterns. Outputs $\boldsymbol{O}_1$ and $\boldsymbol{O}_2$ are presented to the expert for analysis and validation.

Medical status of each patient's status at successive time points can be expressed as a sequence of clusters. The status of patient p_i at time point v_{ij} is responsible for patient's assignment to cluster c_{ij} at that point. Let Seq_i be a sequence of cluster assignments for patient p_i at time points $v_{i1}, v_{i2}, ..., v_{ik_i}$, denoted as $Seq_i = (c_{i1}, c_{i2}, ..., c_{ik_i}) \subseteq C$, where k_i is the number of visits to the clinician by patient p_i and $C = \{c_1, c_2, ..., c_K\}$ is the clustering (set of cluster labels) on the symptoms data. We denote the set of all cluster switching sequences for all patients as Seq

$$Seq = \bigcup_{1 \leq i \leq p} Seq_i, \tag{7}$$

where p denotes the number of analyzed Parkinson's disease patients.

The approach is inspired by natural language processing (NLP) approaches. In NLP, an n-gram is defined as a contiguous sequence of n items from a given sequence of text or speech. In order to analyze the patterns of symptoms changes across all the patients, we perform skip-gram analysis on Seq. A patient's sequence Seq_i can be regarded as an individual document in corpus Seq, where each cluster assignment c_{ij} represents the j-th word in document Seq_i.

The definition of k-skip-n-grams (Guthrie et al. 2006) for a document constructed from words $w_1 ... w_l$ can be expressed as

$$\{w_{i_1}, w_{i_2}, ..., w_{i_n} | \sum_{1 \leq j \leq n} i_j - i_{j-1} < k\} \tag{8}$$

Skip-grams reported for a certain skip distance k allow a total of k or less skips to construct the n-gram. Thus, 3-skip-n-gram results include 3 skips, 2 skips, 1 skip, and 0 skips. The 0-skip-n-grams are n-grams formed from adjacent words. The algorithmic construction of k-skip-n-grams starts with unigrams (which are 0-skip-1-grams) and progressively increases both the skip and the length of the sequence until the required k and n are reached.

We use skip-grams to determine the most frequent statuses of patients, and the most frequent patterns of their symptoms changes. Using skip-grams, the number of investigated n-grams significantly increases, thus providing more reliable introspection into cluster crossings. By skipping certain time points, we take into account that patients' statuses may occasionally result from other factors rather than the natural progression of the disease or the medication therapy. For example, the patient's non-motor symptoms (i.e. depression, apathy, etc.) may worsen due to a sudden death in the family, loss of a friend or loss of a pet. In other words, skip-grams make the resulting patterns more robust compared to the n-grams.

We present an example illustrating the advantage of using skip-grams instead of n-grams in analyzes of sequences. Lets say that a Parkinson's disease patient (p_i) has had 5 visits to the clinician. Based on the patient's symptoms, in each visit the patient was assigned to the following clusters $Seq_i = (1, 0, 2, 0, 1)$ (on visit 1, the patient was assigned to the cluster labeled as *1*, on the second visit, the patient was assigned to the cluster with label *0*, etc.).

The sets of sequences obtained for *bigrams*, *2-skip-bigrams*, *trigrams*, and *2-skip-trigrams* are presented below:

bigrams: {10, 02, 20, 01}
2-skip-bigrams: {10, 12, 10, 02, 00, 01, 20, 21, 01}
trigrams: {102, 020, 201}
2-skip-trigrams: {102, 100, 101, 120, 121, 101, 020, 021, 001, 201}

Using skip-grams to identify interesting patterns in short series of disease progression (reflected by cluster changes) is novel and we are not aware of other equally effective and noise-tolerant method for analysis of really short series. Another seemingly related approach would be to compute frequent itemsets used with association rules (Agrawal et al. 1993) but note that itemsets do not preserve temporal aspect of sequences which is an important information for disease progression.

5 Results of data analysis

The experimental work of this paper is divided into four parts. Initially, we are interested in whether Parkinson's disease patients can be divided into groups of patients with similar symptoms. After determining the appropriate number of clusters in Section 5.1, we report results of two experimental settings: i) using k-means clustering of the merged symptoms data set, and ii) using multi-view clustering on seven separate data sets (seven separate views). The results of both clustering experiments are presented in Section 5.2. This analysis was followed by an attempt to understand the effects of medications changes on the changes of patients' symptoms; these results are presented in Section 5.3. Finally, in Section 5.4 we present the results of experiments intended to find patterns in Parkinson's disease progression. The four groups of reported results were obtained with methodology described in Sections 4.1, 4.2, 4.3, and 4.4, respectively.

5.1 Determining the number of clusters

In order to determine the optimum number of clusters we ran the k-means clustering algorithm on the merged data set using different values for k. The obtained clusters were evaluated using the cluster validity indices introduced in Section 4.1. In terms of these cluster validity indices, better clustering quality is indicated by larger values of SA and CH indexes and lower values of DB index (Liu et al. 2010).

The results of k-means clustering presented in Table 3 show scores obtained for different values of k. The table indicates that k-means clustering produces the best clusters when the value of k is set to 2 or 3. The clustering quality decreases for $k > 3$, as indicated by all of the considered cluster validity indices.

We hypothesize that the reason for no difference in DB and CH indexes when $k = 2$ and $k = 3$, while there is a significant difference in SA, is due to differences how these indices are computed: DB and CH compare distances to centroids, while SA uses nearest neighbors between the instances.

Setting the value of k to 2 would divide patients into two groups: one with a good status and the other with a bad status of PD symptoms; this grouping would not take into account other values of symptoms except the ones characterized as either normal or very problematic for the patients. For this reason and to provide more variability we set the value of k to 3 to get three patient clusters instead of just two.

Table 3 Values of clustering validity scores for different number of clusters. Clusters are generated on the merged data set, using the *k*-means clustering algorithm

Number of clusters	Silhouette index (SA)	Davies-Bouldin index (DB)	Calinski-Harabasz index (CH)
2	0.516	0.916	162.540
3	0.347	0.916	162.540
4	0.371	1.266	54.103
5	0.279	1.133	40.545
6	0.276	1.458	32.412
7	0.258	1.489	26.992

Please note that the selection of k-means approach for clustering the merged data set was done after series of experiments. We checked three clustering approaches: k-means, k-medoids (Kaufman and Rousseeuw 1987), and DBSCAN (Ester et al. 1996). For each of the considered approaches, we evaluated the produced clusters using the validity indices SA (Rousseeuw 1987), DB (Davies and Bouldin 1979), and CH (Caliński and Harabasz 1974). Based on the results, we decided to use k-means as our clustering method of choice. Evaluation details for the considered clustering approaches can be found in Appendix A in Table 5.

5.2 Results of symptoms data analysis

To determine the progression of patients' symptoms, for each Parkinson's disease patient from our data set and for each two consecutive time points we investigated changes of clusters in which the patient participated. With the help of the expert, we order the clusters according to the quality of life indicators (i.e. severity of symptoms) of patients in the clusters. The evaluation of the quality of discovered clusters is two-fold. Clusters are initially evaluated using the internal cluster validity indices: SA, DB, and CH. The generated clusters are described by rules produced with the DoubleBeam-RL algorithm, and these descriptions are presented to experts. Based on the rules, experts order clusters according to the severity of symptoms of patients involved in each cluster.

5.2.1 Results of merged symptoms data analysis

The classification rules describing the clusters obtained from the merged symptoms data analysis are presented in Table 4. The rules indicate that the clusters are linearly ordered (with indexes 0, 1, and 2) and contain instances (patients symptoms recorded at a certain time point) with different severity of their motor symptoms. *Cluster 0* consists of instances with the sum of motor symptoms severity up to 22 (out of 138). Patients that have slightly worse motor symptoms are assigned to *Cluster 1* (sum of motor symptoms severity between 23 and 42). In *Cluster 2* there are patients whose motor symptoms significantly affect their motor functions (sum of motor symptoms severity greater than 42). The worsening of motor symptoms is followed by the aggravation of non-motor symptoms, mostly autonomic symptoms (sleeping, urinary, or constipation problems). This can be observed by the increased values of attributes SCAUSUM and NP2SUM in the rule sets describing *Cluster 1* and *Cluster 2*.

Inspection of the time line of cluster changes for a single patient. In order to illustrate the cluster changes for a patient, we subjectively chose a patient who already completed

Table 4 Rules describing clusters obtained by k-means clustering on the concatenated data set of attribute sums. Variables p and n denote the number of covered true positive and false positive examples respectively. We present the complete rules generated by the DoubleBeam-RL algorithm which does not prune its learned rules

Rule		p	n
Rules for cluster 0			
NP3SUM $\leq$ 20	$\rightarrow$ cluster = 0	488	4
NP3SUM $\leq$ 21 AND NP2SUM $\leq$ 6	$\rightarrow$ cluster = 0	321	0
NP3SUM = (19, 22] AND NP1SUM = 0	$\rightarrow$ cluster = 0	54	23
Rules for cluster 1			
NP3SUM = (22,30]	$\rightarrow$ cluster = 1	323	13
NP3SUM = (30, 39] AND SCAUSUM = (4, 10]	$\rightarrow$ cluster = 1	91	17
NP3SUM = (22, 42] AND NP2SUM = (0, 6]	$\rightarrow$ cluster = 1	206	6
NP3SUM = (22, 34] AND SCAUSUM = (10, 17] AND PASESUM >9	$\rightarrow$ cluster = 1	101	6
Rules for cluster 2			
NP3SUM >42	$\rightarrow$ cluster = 2	125	1
NP3SUM >37 AND NP1PSUM >5 AND MCAVFNUM $\leq$ 18	$\rightarrow$ cluster = 2	123	6
NP3SUM >30 AND NP2SUM >17	$\rightarrow$ cluster = 2	82	0
SCAUSUM >20 AND NP2SUM >9 AND MCAVFNUM $\leq$ 24	$\rightarrow$ cluster = 2	54	18
NP3SUM >30 AND SCAUSUM >11 AND NP2SUM >12	$\rightarrow$ cluster = 2	123	2
NP3SUM >36 AND SCAUSUM >6 AND NP2SUM >6 AND NP1PSUM >2	$\rightarrow$ cluster = 2	168	6

involvement in the PPMI study, and present her changes in the overall status in Fig. 2. The disease status can be tracked through changes in the patient's cluster assignments recorded during consecutive visits to the clinician. We also present the changes in medications therapy, made in order to keep the patient's symptoms as stable as possible.

We presented the figure (as well as the symptoms and medications data) to our consulting clinician for interpretation. He commented that the particular treatment was in accordance with the standard practice and guidelines for the treatment of Parkinson's disease patients. The usual practice is that clinicians almost always start with MAO-B inhibitors (such as Azilect) to protect neurons and later introduce dopamine agonists (such as Requip or Neupro) in order to manage Parkinson's disease (European Parkinson's Disease Association 2016). The usage of levodopa (Carbidopa/Levodopa) is delayed as long as possible—symptoms allowing—in order to avoid the side effects of prolonged usage of levodopa, such as dyskinesia and on/off fluctuations.

As evident from the diagram, the initial status of the patient was good. The clinician started the treatment of Parkinson's disease by introducing a MAO-B inhibitor (Azilect). Then clinician increased the dosage, trying to find an appropriate dosage for the specific patient. Once the patient's symptoms worsen (as indicated by the cluster changes between visits V04 and V06), the clinician introduced dopamine agonists to stabilize the symptoms. There were several adjustments aiming to find the appropriate dopamine agonist therapy

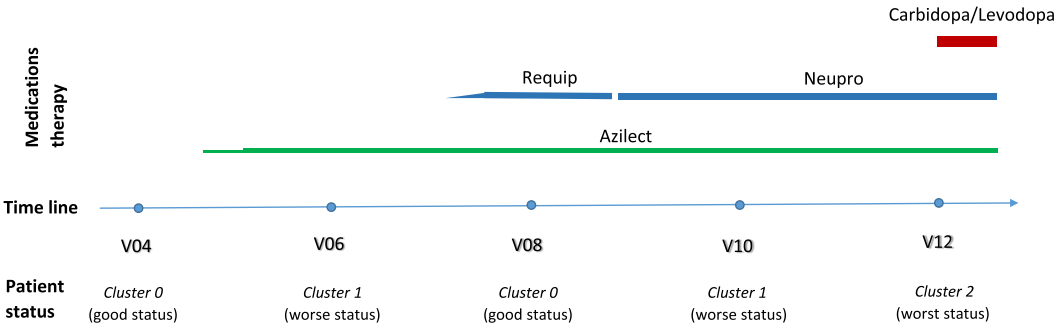


Fig. 2 Inspection of a cluster change time line of a single patient. Points on the time line present visits the patients has made to the clinician. Patient's medications therapy is presented by the groups of antiparkinson medications the patient has received during her involvement in the PPMI study. The color of medications therapy determines the group of antiparkinson medications—MAO-B inhibitors are presented with the green line, dopamine agonists are presented with the blue line, and levodopa based medications are presented with the red line on the top. Line width indicates the value of LEDD, i.e. the ticker the line the higher the value of LEDD. Endpoints of lines indicate beginnings and ends of treatments with particular medications. For example, the patient was treated with Requip (a dopamine agonist medication) starting between visit 6 (V06) and visit 8 (V08) and ending sometimes after visit 8. The treatment with Neupro (with almost the same value of LEDD) started immediately after the treatment with Requip stopped and was ongoing even after the patient finished her involvement in the PPMI study (V12)

for the patient: the clinician started with Requip and changed the medication's dosage several times (represented by the steep increase of the blue line). The patient initially reacted well to this change and her overall status was improved (V08). However, the status then worsened and the clinician changed the therapy by ending the intake of Requip and introducing Neupro. This medication change did not improve the patient's status at visit V10, and by visit V12 her status got even worse (our methodology assigned the patient to *cluster 2* at visit V12). Since the patient's status was bad and the quality of life has significantly declined, the clinician was forced to introduce levodopa.

5.2.2 Results of multi-view symptoms data analysis

In addition to analyzing the merged symptoms data, we performed a number of experiments on multi-view symptoms data consisting of seven separate symptoms data sets. In these experiments, we used the CoNMF multi-view clustering algorithm (He et al. 2014). Similarly to the merged view clustering approach, we tried to compare the clusters obtained by the multi-view approach by the severity of patients' symptoms assigned to them. The analysis has revealed that there were no significant intersections of the instances assigned to the clusters obtained by the multi-view approach compared to the clusters obtained by k-means ($k = 3$) clustering of the merged symptoms data set. Furthermore, given that the distinction between the three produced clusters was unclear, the ordering and comparison of the clusters was not possible This result means that we were not able to interpret the clustering produced by the CoNMF algorithm. In Appendix C we present the results of further analysis on impact different views have in the multi-view clustering process.

Results from Table 8,[3] and Table 10 in Appendix C show that the quality of clusters induced using the CoNMF approach is lower than the quality of clusters generated on

[3]Note that in Table 9 we present the Adjusted Random Index values where we compare the cluster similarity between the three best performing bi-view clustering settings.

the merged data set. Results reveal that it is beneficial to combine multiple data sets in order to obtain better clusters and better overall picture of the patients that were assigned to these clusters. However, when including new views, one must be careful, since the inclusion of seemingly uncorrelated views can hinder the performance of the multi-view approach. Results from Table 10 show that the best quality clusters are obtained when using only three data sets (views): SCOPA-AUT, MDS-UPDRS Part II, and MDS-UPDRS Part III. Due to low quality of induced clusters, we decided not to investigate the changes of medications dosages with respect to the changes of clusters generated in the multi-view clustering setting. However, in future, we will consider also other multi-view clustering algorithms.

Rules discovered with the best multi-view clustering are presented in Tables 11, 12, and 13 in Appendix D. The groups of patients are described mostly by their motor symptoms and descriptions are supported by attributes from the data set SCOPA-AUT. The SCOPA-AUT data set contains information about the autonomic symptoms of patients, namely mostly constipation and urinary problems. Even though the resulting multi-view clustering is of low quality and the experts were not able to order the produced clusters by the severity of the symptoms of patients involved in them, the consulted experts were pleased with the discovery that autonomic symptoms from SCOPA-AUT play an important role in produced clusters, as recent research shows that autonomic symptoms can be a potential premotor marker of Parkinson's disease (Ceravolo et al. 2010).

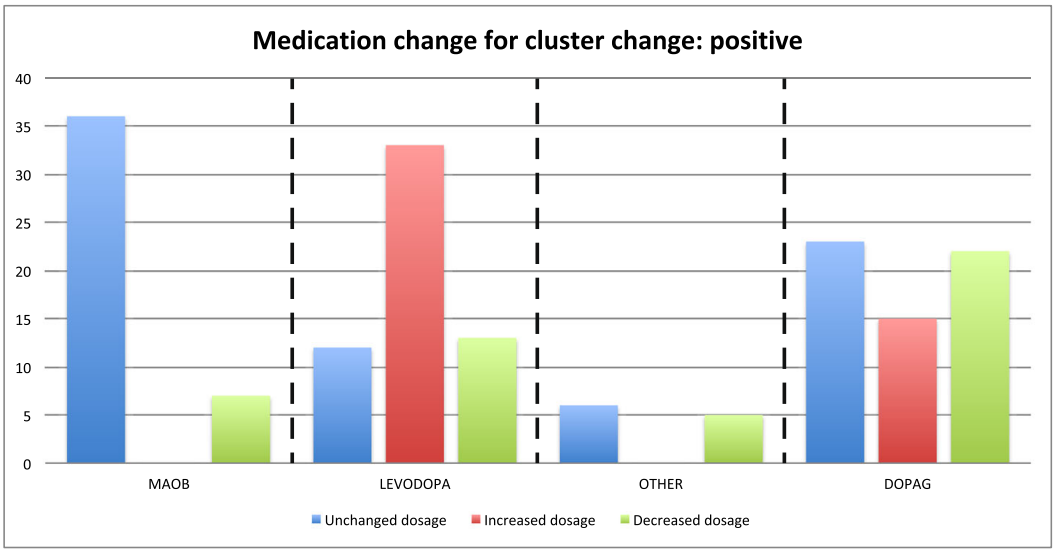
5.3 Results of medications change analysis

The experts were able to order clusters obtained from the merged symptoms data (presented in Table 4) by the severity of symptoms. The order was total (all clusters were comparable), so we assigned the three clusters indexes 0, 1, and 2 (lower index means lower severity of symptoms). When a patient moves from a cluster with a lower index to the one with a higher index, the patient's symptoms have worsened and we consider this change to be negative. A positive cluster change is recorded if the patient's symptoms have improved and the patient moves to a cluster with a lower index. The medications change patterns for positive and negative cluster change were obtained with the approach described in Section 4.3. The results are shown in Fig. 3.

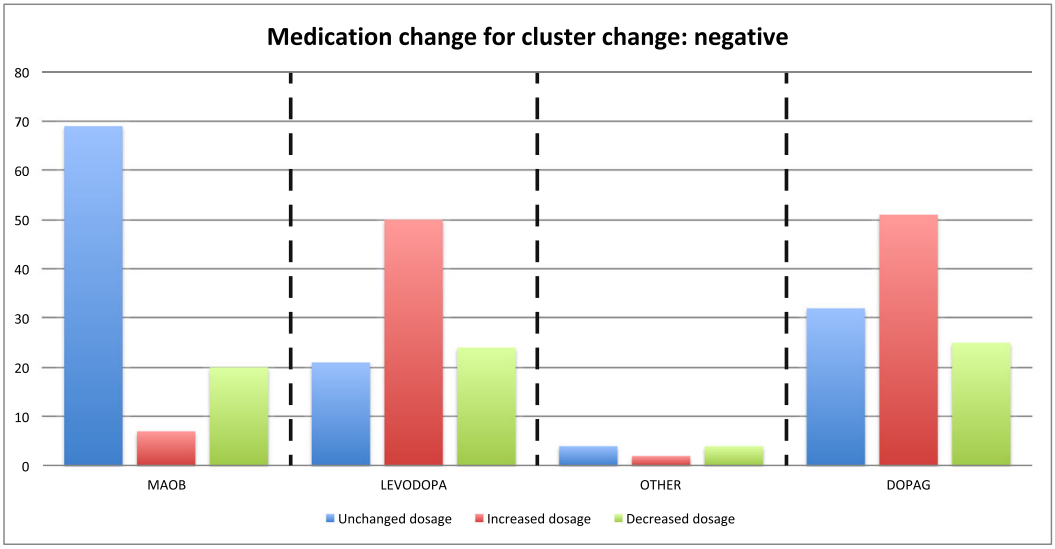
Figure 3a shows the medications changes when a positive cluster change has occurred. The red bars represent the number of times the dosage of medications from certain medication group has increased. Similarly, the number of times the medication dosage has decreased is shown in green. Blue bars present the number of times when a positive cluster change has occurred, but the medication dosage has stayed unchanged.

Figure 3b outlines the medications changes when a negative cluster change has taken place. These two graphs show patterns of medications modifications as a result of significant changes in the patient's status (patient's symptoms in two consecutive time points changed significantly, thus prompting a cluster change).

Figure 3 indicates that the patients' motor symptoms improve when the dosage of medications from the levodopa drug group is increased and the dosage of dopamine agonists is decreased or stays the same. When the dosage of both levodopa medications and dopamine agonists is increased the motor symptoms of the patients worsen. Clinicians prescribe and gradually increase the dosages of levodopa to handle the motor symptoms of patients. The usage of high dosages of dopamine agonists produces side effects affecting the non-motor



(a) Positive cluster change.



(b) Negative cluster change.

Fig. 3 Recorded Parkinson's disease medications changes when patient's cluster allocation has changed. Clusters were obtained from merged symptoms data set. A positive cluster change indicates that the patient's symptoms improved. A negative cluster change occurs when the patient's symptoms worsen. Medication groups are visually divided by vertical dashed lines

symptoms of the patients. A decrease of dosage eliminates these side effects and improves the patient's status.

5.4 Disease progression patterns

Figure 4 presents the results from the 3-skip-2-gram analysis of cluster crossings in the merged symptoms clustering setting. The results indicate that the patients' status is mostly stable over the considered time points. Patients tend to stay in the clusters they were initially assigned to. This is followed by a portion of patients whose symptoms worsen

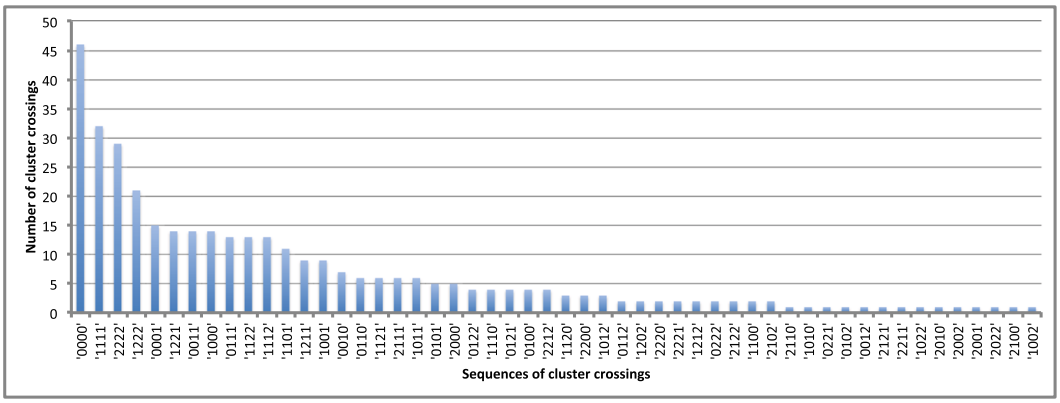


Fig. 4 Histogram resulting from 3-skip-2-gram analysis. The possible cluster crossings are listed on the X-axis (e.g., 01 indicates that a patient has moved from *Cluster 0* to *Cluster 1*), while the Y-axis represents the number of cluster crossings

(cluster crossings 01 and 12) and those whose symptoms improve (cluster crossings 10 and 21). These symptoms changes have all occurred gradually—patients have moved to the adjacent cluster. The number of patients whose symptoms have significantly changed (cluster crossings 02 and 20) is much lower.

The analysis of bigrams (2-grams) in Fig. 4 cannot reveal trends in patients' status over longer time period. Figure 5 presents the patterns of 4 almost consecutive cluster crossings obtained by 3-skip-4-gram analysis of the sequences of cluster crossings on the merged data set. It confirms the results from Fig. 4 which indicate that patients' status is usually stable and they tend to stay in the same cluster to which they were initially assigned.

Figure 5 reveals existence of interesting and slightly unexpected patterns of symptoms change: 1001, 0110, and 2000. We selected these sequences (subjectively) as patients' conditions are not steadily deteriorating and use them to illustrate our approach—the patients with similar symptoms have similar patterns of disease progression. We discuss groups of patients with 0110 and 2000 pattern below.

The analysis of patients with the 0110 cluster change pattern reveals that these are younger patients (50–64 years old) who were enrolled in the PPMI study soon after

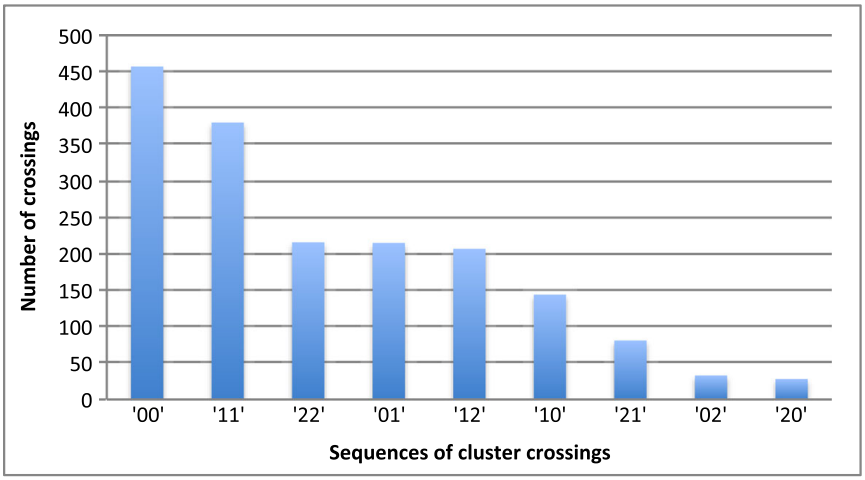


Fig. 5 Histogram resulting from 3-skip-4-gram analysis. The possible cluster crossings are listed on the X-axis and the Y-axis represents the number of cluster crossings

their Parkinson's disease diagnosis (in less than 6 months). A common thread of these patients is that they have had problems with anxiety at some point of the disease (quantified with score 1 – a symptom is present and has a slight influence on the patient's quality of life). Most of these patients have also started feeling a decline in their cognitive functions. These patients were treated with the combinations of dopamine agonists and MAO-B inhibitors. When patients motor symptoms have slightly worsened, the clinicians have tried to stabilize them by increasing the dosage of dopamine agonists, changing the dopamine agonists medication, or in rare cases introducing levodopa. These treatments are in accordance with the new practices for Parkinson's disease—clinicians introduce MAO-B inhibitors to protect the neural system of the patient, and prescribe dopamine agonists in order to control motor symptoms that are bothering the patients, and in that way they prolong the time before levodopa is introduced in the therapy.

An inspection of patients with cluster change pattern 2000 reveals that two patients who exhibit this pattern are elder female patients (more than 71 years old), with two years between the time of their diagnosis and their enrolment into the PPMI study. In the time of their initial visits, both patients had problems with their facial expression, problems with fingerntapping, hand movement, pronation-supination, toe-tapping, leg agility, postural tremor, rest tremor amplitude and constancy of rest tremor. For both patients, these symptoms were prominent on the left-hand side. In addition to their motor problems, both patients have experienced problems with depression and anxiety. Patients' medications log reveals that once the patients' motor symptoms were deemed problematic (at that time point the patients were assigned to *Cluster 2*), their respective clinicians started the symptoms treatment with levodopa medications. The introduction of levodopa lead to stabilization of the symptoms, and in our research, we observe a crossing of the patients from *Cluster 2* to *Cluster 0*.

6 Conclusions

The aim of our research is to develop a methodology which will make suggestions to the clinicians about the possible treatment changes that will improve the patient's quality of life. We also aim to discover groups of patients that follow interesting patterns of symptoms change in hope that their disease progression will reveal common symptoms and medications threads, which could benefit the future patients. Our methodology contains tracking the changes in medication patterns, clustering, rule learning and skip-grams. The results confirm known facts about the Parkinson's disease: the motor symptoms, tremor, shaking, involuntary movement, etc. are the characteristic symptoms of the disease and significantly affect the quality of life of the suffering patient. We show that Parkinson's disease patients can be divided into clusters ordered in accordance with the severity of their symptoms. By following the evolution of symptoms for each patient separately, we were able to determine patterns of medications change which can lead to the improvement or worsening of the patients' quality of life.

We introduced skip-grams as a method for following the progression of the disease. The analysis showed that the progression of the disease is mostly steady in the period of five years involvement in the PPMI study—the patients stay in the initially assigned clusters or they move to the adjacent clusters. Analysis of 3-skip-4-grams outlined groups of patients

with interesting patterns of cluster changes. We detected a group of older patients, who were not treated for a longer period and whose treatment consists of direct introduction of levodopa for treatment of motor symptoms. The other interesting group are younger patients, who were recently diagnosed with Parkinson's disease and whose treatment included the combinations of MAO-B inhibitors and dopamine agonists. In further work, we will consult medical experts for specific patients with interesting sequences and ask them to interpret their etymological characteristics, motor symptoms, and changes of therapy.

Results from the multi-view clustering setting are underwhelming in terms of the quality of produced clusters. However, the results reveal the importance of autonomic symptoms to the quality of life of Parkinson's disease patients.

The rules describing the obtained clusters were either very general (merged view setting) or very specific (multi-view setting) and may not be of sufficient assistance to clinicians. This is due to the nature of the used data, i.e. a vector of attribute sums (merged view) or a high-dimensional vector of attributes with numeric values. In future work, we will test our methodology with only a handful of carefully chosen attributes. These attributes, selected with the help of Parkinson's disease specialists, will be described by nominal values used in the clinicians' everyday practice i.e. normal, non-problematic, problematic. We believe that by an expert-assisted decrease of feature space dimensionality, we will be able to obtain descriptions of groups of patients which are even more meaningful and helpful to the clinicians. Additionally, we will improve the medications suggestion process to produce numerical suggestions of medications dosages which should be prescribed to the patients. An interesting direction for further work is to explore other clustering approaches, in particularly hierarchical clustering. Attributes from the MDS-UPDRS and MoCA questionnaires can be ordered hierarchically and exploiting this characteristic may lead to better defined groups of patients with similar symptoms. Transitions between such clusters could reveal more specific and detailed patterns of disease progression. Besides skip-grams we plan to explore other possibilities to handle temporal data. For example, we want to compare the state of a patient in a given time point with all of its past time points (not only the immediately preceding one).

Acknowledgements This work was supported by the PD_manager and HBP SGA1 projects, funded within the EU Framework Program for Research and Innovation Horizon 2020 grants 643706 and 720270, respectively. We acknowledge also the support of the Slovenian Research Agency (research core funding P2-0103 and P2-0209).

Data used in the preparation of this article were obtained from the Parkinson's Progression Markers Initiative (PPMI) (www.ppmi-info.org/data). For up-to-date information on the study, visit www.ppmi-info.org. PPMI—a public-private partnership—is funded by the Michael J. Fox Foundation for Parkinson's Research and funding partners. Corporate Funding Partners: AbbVie, Avid Radiopharmaceuticals, Biogen, BioLegend, Bristol-Myers Squibb, GE Healthcare, GLAXOSMITHKLINE (GSK), Eli Lilly and Company, Lundbeck, Merck, Meso Scale Discovery (MSD), Pfizer Inc, Piramal Imaging, Roche, Sanofi Genzyme, Servier, Takeda, Teva, UCB. Philanthropic Funding Partners: Golub Capital. List of funding partners can be also found at www.ppmi-info.org/fundingpartners.

Appendix A: Comparison of clustering algorithms on merged data set

We considered three clustering approaches for the merged data set: k-means, k-medoids, and DBSCAN. We clustered the merged data into different number of clusters and evaluated the quality of the produced clusters with the internal cluster validity metrics: SA (Rousseeuw

Table 5 Cluster validation measures for k-means, k-medoids, and DBSCAN, where k presents the number of clusters. Clustering was performed on the merged data set. Better clusters quality is marked with higher values of SA and CH, and lower values of DB

	k-means			k-medoids			DBSCAN		
k	SA	DB	CH	SA	DB	CH	SA	DB	CH
2	0.516	0.916	162.540	0.505	0.918	162.539	-0.362	0.996	16.946
3	0.368	0.916	162.540	0.336	0.918	162.539	-0.132	0.996	16.946
4	0.371	1.263	54.103	0.318	1.387	54.099	0.250	0.796	297.712
5	0.287	1.151	40.546	0.259	1.235	40.546	nan	nan	inf
6	0.275	1.256	32.412	0.253	1.283	32.412	nan	nan	inf
7	0.284	1.619	26.991	0.253	1.364	26.990	nan	nan	inf

1987), DB (Davies and Bouldin 1979), and CH (Caliński and Harabasz 1974). Table 5 presents the results of cluster validation for the selected clustering methods and the chosen number of clusters. The results show that the best performing approach is k-means.

Appendix B: Features selected by unsupervised feature selection

We used unsupervised feature subset selection to select the most relevant attributes for clustering algorithms. We used the SPEC algorithm (Zhao and Liu 2007) implemented in

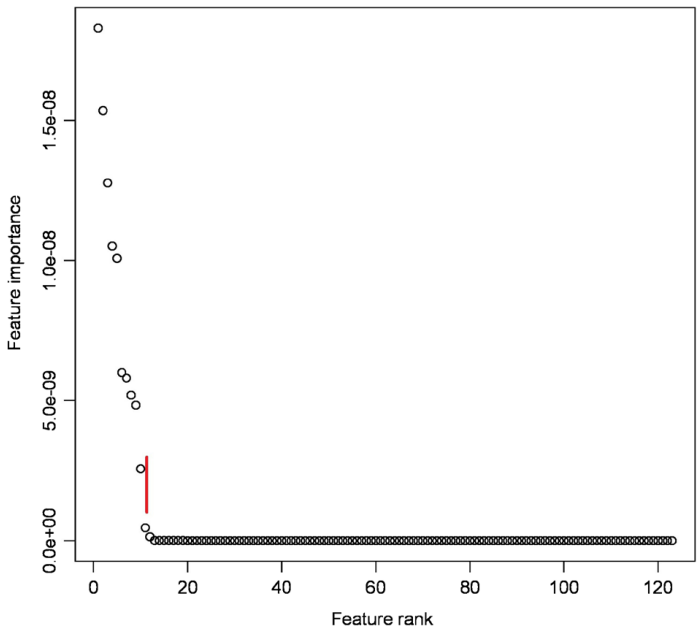


Fig. 6 Attribute rank vs attribute importance as determined by the SPEC algorithm (the most influential attribute has rank 1)

Table 6 The most important attributes ordered according to SPEC (see Fig. 6)

Attribute	Attribute description	Data set
MCAREC4	Delayed recall - daisy	MoCA
NHY	Hoehn and Yahr score	MDS-UPDRS Part III
NP3PTRML	Postural tremor (left hand)	MDS-UPDRS Part III
NP3SPCH	Speech problems	MDS-UPDRS Part III
NP2EAT	Eating tasks	MDS-UPDRS Part II
NP1SLPD	Daytime sleepiness	MDS-UPDRS Part Ip
NP3RIGLU	Rigidity (left arm)	MDS-UPDRS Part III
NP1PAIN	Pain and other sensations	MDS-UPDRS Part Ip
NP3FTAPL	Finger tapping (left hand)	MDS-UPDRS Part III
NP3RTCON	Constancy of rest	MDS-UPDRS Part III

Python (Li et al. 2016). Figure 6 presents the evaluation of attributes relevance. Based on the results, we selected the attributes left from the red line in Fig. 6. This resulted in a list of 10 attributes, presented in detail in Table 6.

In Table 7 we present the cluster validation values on the data set containing only the best attributes (listed in Table 6). The results reveal that the merged data set (consisting of sums of attributes) produces better quality clusters than the data set reduced with feature subset selection.

Results from Tables 5 and 7 show that better clusters are produced when sums of attribute values from the considered views are used as attributes in the merged data set. Parkinson's disease patients experience a whole range of symptoms, both motor and non-motor, and it is tougher for traditional clustering algorithms to separate them into groups of similar patients. The introduction of sums makes it possible to have a view of the overall status of the patients concerning particular sets of symptoms (i.e. motor symptoms, non-motor symptoms, autonomic symtptoms etc.).

Table 7 Cluster validation measures for k-means, k-medoids, and DBSCAN, where k presents the number of clusters. Clustering was performed on the data set containing only attributes from Table 6

	k-means			k-medoids			DBSCAN		
k	SA	DB	CH	SA	DB	CH	SA	DB	CH
2	0.379	1.199	102.657	0.379	1.199	102.657	0.379	1.199	102.657
3	0.337	1.199	102.657	0.283	1.199	102.657	nan	1.199	102.657
4	0.296	1.590	34.168	0.217	1.781	34.168	nan	nan	inf
5	0.279	1.580	25.608	0.170	1.745	25.607	nan	nan	inf
6	0.267	1.617	20.471	0.189	1.649	20.471	nan	nan	inf
7	0.262	1.694	17.046	0.182	1.969	17.047	nan	nan	inf

Appendix C: Evaluation of multi-view clusterings

In order to determine how the choice of data sets influence the results of multi-view clustering, we executed multi-view clustering on all 21 pairs of views, i.e. $\frac{7 \cdot 6}{2}$ pairs. Clusters resulting from each pair were evaluated using SA (Rousseeuw 1987) and the results are presented in Table 8. SA is a normalized value (range from -1 to 1) and is used to compare cluster quality on these data sets. Since clustering was performed on different data sets (each pair is effectively a different data set) and values of DB and CH are not comparable across data sets, we do not present these values. The value of each cell in Table 8 corresponds to the quality of clusters obtained by multi-view clustering on the data sets from the corresponding row and column. For example, SA (Rousseeuw 1987) on clusters obtained by multi-view clustering on the MDS-UPDRS Part I (NUPDRS1) and MoCA is 0.021. The best cluster is marked with bold.

The results show that all pairs produce clusters with low quality, but the three best performing pairs according to SA are: (SCOPA-AUT, MDS-UPDRS Part II), (MDS-UPDRS Part III, MDS-UPDRS Part II), and (PASE, MDS-UPDRS Part II).

We used the Adjusted Rand Index (ARI) (Hubert and Arabie 1985) to compare cluster structures discovered by different cluster configurations. The value of ARI is 0 for two random clusterings and 1 for two identical clusterings. Table 9 presents the ARI score computed on pairs of the winning two-view clustering settings. Results reveal that all pairs of clusterings are quite similar, and the (NUPDRS3, NUPDRS2P) and (PASE, NUPDRS2P) pairs produce almost identical clusters ($ARI = 0.966$). As the quality of individual pairs is rather low (see Table 8), there is little chance that further combinations of views would improve the quality.

Nevertheless, we constructed two additional settings for multi-view clustering by systematically adding views (data sets) to the winning bi-view clustering setting (SCOPA-AUT, MDS-UPDRS Part III). We in turn added the remaining data sets from the second (MDS-UPDRS Part II and MDS-UPDRS Part III) and third (PASE and MDS-UPDRS Part III) best performing bi-view clustering setting, thus obtaining two new multi-view settings: (SCOPA-AUT, MDS-UPDRS Part II, MDS-UPDRS Part III) and (SCOPA-AUT, MDS-UPDRS Part II, MDS-UPDRS Part III, PASE). We evaluated the quality of clusters produced by these three settings and presented the results in Table 10, where we also

Table 8 Value of SA on clusters discovered with multi-view clustering on pairs of data sets. Higher values of SA indicate clusters with better quality

	MOCA	NUPDRS1	NUPDRS1P	NUPDRS2P	NUPDRS3	PASE
NUPDRS1	0.021					
NUPDRS1P	0.023	0.014				
NUPDRS2P	0.022	0.033	0.024			
NUPDRS3	0.025	0.038	0.015	0.168		
PASE	0.023	0.059	0.013	0.162	0.048	
SCOPA-AUT	0.024	0.018	0.013	**0.173**	0.047	0.031

Table 9 ARI scores for the best performing pairs of two-view multi-view clusterings

	(NUPDRS3, NUPDRS2P)	(PASE, NUPDRS2P)
(SCOPA, NUPDRS2P)	0.488	0.504
(NUPDRS3, NUPDRS2P)		0.966

included the cluster quality measures when all views are considered and the scores of the best single view clustering on the merged data set. Please note that since clustering was performed on different data sets, values of DB and CH are not comparable. SA is a normalized value (range from -1 to 1) and is used to compare cluster quality on these data sets.

Based on the SA values from Table 10, clustering with the best clustering is produced on the merged data set that consists only of sums of attribute values from 7 data sets from Section 3.3. In the multi-view setting, best results were obtained when three data sets were considered (SCOPA-AUT, MDS-UPDRAS Part II, MDS-UPDRS Part III). The SCOPA-AUT data set contains attributes describing the autonomic symptoms of patients. The MDS-UPDRS Part II data expresses 'motor experiences of daily living', including speech problems, the need for assistance with the daily routines such as eating or dressing, etc, while the MDS-UPDRS Part III data set describes the motor symptoms which are the most characteristic symptoms of Parkinson's disease. Even though the clusters produced by the multi-view setting are of lower quality than those produced on the merged data set, results from Table 10 reveal that it might be beneficial to combine multiple data sets: the inclusion of the MDS-UPDRS Part III data set in the best performing bi-view clustering setting (SCOPA-AUT, MDS-UPDRS Part II) (SA = 0.173) produces clusters with an improved quality (SA = 0.205). These results also show that the inclusion of other, seemingly uncorrelated data sets (PASE, MOCA, MDS-UPDRS Part I, MDS-UPDRS Part Ip) can lead toward significant decrease in the quality of clusters.

In addition to the work presented above, we also used unsupervised feature subset selection to select the most relevant attributes from each of the seven views (data sets). We evaluated the quality of clusters on the newly generated data sets following the procedure presented in this section. Results showed that the quality of the clusters in these new set-

Table 10 Comparison of cluster quality using silhouette analysis (SA) for different setting of multi-view clustering

The *Data set* column presents the symptoms data sets that are used in the multi-view clustering

Data set	SA
SCOPA, NUPDRS2P	0.173
SCOPA, NUPDRS2P, NUPDRS3	0.205
SCOPA, NUPDRS2P, NUPDRS3, PASE	0.0195
All 7 data sets	0.0514
Merged data set	0.347

tings was significantly lower than the quality of clusters presented here. For that reason we did not include this part of research into the paper.

Appendix D : Rules describing multi-view clusters

We present rules describing clusters obtained by multi-view clustering using three views (SCOPA-AUT, MDS-UPDRS Part II, and MDS-UPDRS Part III) i.e. the best multi-view clustering according to SA from Table 10. Attributes with the prefix SCAU are symptoms from the SCOPA-AUT data set. The suffix in the names of these attributes designates the nature of the autonomic symptoms. Attributes SCAU1-SCAU7 describe gastrointestinal symptoms, urinary problems are recorded by attributes SCAU8-SCAU13, while attributes SCAU14-SCAU16 hold information about patient's cardiovascular problems. Attributes SCAU17-SCAU18, SCAU20-SCAU21 describe thermoregulatory problems, while attribute SCAU19 describes any pupillomotor issues that a patient might be experiencing. Attribute prefixes determine the data set of their origin. Attributes with prefix NP2 are from the MDS-UPDRS Part II, while the prefix NP3 designates attributes from the MDS-UPDRS Part III data set (including attributes NHY and DYSKPRES).

Tables 11, 12, and 13 present rules describing *cluster 0*, *cluster 1*, and *cluster 2* respectively, obtained by multi-view clustering. Rules are induced on the data set that is a concatenation of the three views: SCOPA-AUT, MDS-UPDRS Part II, and MDS-UPDRS Part III. Contrary to the rules obtained by the single view clustering on the merged data set where groups of patients were described by the severity of their overall status, the multi-view clusters are described by symptoms. These rules mostly describe the motor status of Parkinson's disease patients (attributes from MDS-UPDRS Part III), and are supported by their motor ability in daily living (attributes from MDS-UPDRS Part II) and their autonomic symptoms (SCOPA-AUT).

Table 11 Description rules for *cluster 0* of the multi-view clustering approach generating clusters with best quality. Views were represented by the SCOPA-AUT, MDS-UPDRS Part II, and MDS-UPDRS Part III data sets

Rule		p	n
IF:			
SCAU19 $\leq$ 2 AND NP3PSTBL $\leq$ 2 AND NP3HMOVL $>$ 2 AND			
NP3GAIT $\leq$ 1 AND NP3RTARU $\leq$ 0	$\leftarrow$ cluster $= 0$	58	0
ELSE IF:			
NP3RISNG $\leq$ 1 AND NP3FTAPR $\leq$ 0 AND NP3HMOVL $>$ 0 AND			
NP3HMOVR $\leq$ 0 AND NP3RTARU $\leq$ 0 AND NP3FACXP $\leq$ 2	$\leftarrow$ cluster $= 0$	146	8
ELSE IF:			
NP3FTAPL $>$ 1 AND NP2FREZ $\leq$ 0 AND NP3RTARU $\leq$ 0 AND			
NP3RTALU $>$ 0	$\leftarrow$ cluster $= 0$	87	0
ELSE IF:			
NP2SALV $\leq$ 3 AND NP3FRZGT $\leq$ 0 AND NP3FTAPL $>$ 2 AND			
NP3LGAGL $>$ 0	$\leftarrow$ cluster $= 0$	25	4

Table 12 Description rules for *cluster 1* of the multi-view clustering approach generating clusters with best quality. Views were represented by the SCOPA-AUT, MDS-UPDRS Part II, and MDS-UPDRS Part III data sets

Rule		p	n
IF:			
NP3RIGLU ≤ 0 AND NP3RIGN ≤ 1 AND NP3RTARU > 1	$\leftarrow$ cluster $= 1$	143	3
ELSE IF:			
NP3RTCON > 2 AND SCAU18 ≤ 1 AND NP2SUM ≤ 15 AND			
NP3RTARU > 1 AND NP3FACXP ≤ 2	$\leftarrow$ cluster $= 1$	83	3
ELSE IF:			
NP3RIGLL ≤ 0 AND NHY ≤ 1 AND NP3RTARU > 0 AND			
SCAU6 ≤ 1	$\leftarrow$ cluster $= 1$	40	6
ELSE IF:			
NP3PRSPL ≤ 0 AND NP3RTCON > 1 AND NP3RTARU > 0	$\leftarrow$ cluster $= 1$	37	2
ELSE IF:			
SCAU12 ≤ 1 AND NP3RTALL ≤ 0 AND NP2TRMR > 0 AND			
NP2EAT ≤ 0 AND NP3HMOVL ≤ 0 AND SCAU20 ≤ 0 AND			
NP3RTARU > 0 AND SCAU7 ≤ 0	$\leftarrow$ cluster $= 1$	15	3
ELSE IF:			
NP3RIGLU $=$ (0,1] AND NP3RTCON > 1 AND SCAU17 ≤ 1 AND			
NHY ≤ 2 AND NP3RTARU > 1	$\leftarrow$ cluster $= 1$	19	4
ELSE IF:			
NP3RTCON > 2 AND NP2HWRT > 0 AND NP3LGAGL ≤ 0 AND			
NP3TTAPL $=$ (0,1] AND SCAU6 ≤ 1	$\leftarrow$ cluster $= 1$	7	2
ELSE IF:			
NP2SALV ≤ 0 AND NP3RIGLU ≤ 0 AND SCAU17 ≤ 1 AND			
NP2WALK ≤ 0 AND NP3RTARL > 0	$\leftarrow$ cluster $= 1$	5	2
ELSE IF:			
NP2EAT > 0 AND NP3GAIT ≤ 0 AND NP3RTARU > 0 AND			
NP2SPCH ≤ 0 AND SCAU4 ≤ 1	$\leftarrow$ cluster $= 1$	6	1
ELSE IF:			
SCAU18 > 0 AND NP2DRES > 0 AND NP3SPCH ≤ 0 AND			
NP3RTARU > 1 AND NP3RTALU ≤ 0	$\leftarrow$ cluster $= 1$	4	0

Table 13 Description rules for *cluster 2* of the multi-view clustering approach generating clusters with best quality. Views were represented by the SCOPA-AUT, MDS-UPDRS Part II, and MDS-UPDRS Part III data sets

Rule		p	n
IF:			
SCAUSUM > 5 AND NP3RTCON ≤ 0 AND NP3FTAPR > 2	← cluster = 2	36	1
ELSE IF:			
NP3RTCON ≤ 0 AND NP2TRMR ≤ 0 AND NP3TTAPL ≤ 0	← cluster = 2	53	1
ELSE IF:			
NP3RTCON ≤ 0 AND NP3HMOVR > 0 AND NP3TTAPL ≤ 0	← cluster = 2	52	0
ELSE IF:			
NP3RTCON ≤ 0 AND NP3PTRML ≤ 0 AND NP3TTAPR > 1	← cluster = 2	73	12
ELSE IF:			
NP3RTCON = (0,1] AND NP3HMOVR > 2 AND NP3TTAPR > 0	← cluster = 2	15	1
ELSE IF:			
NP3RTALJ ≤ 0 AND NP3GAIT ≤ 1 AND SCAU8 > 2 AND NP3RTALU ≤ 0 AND SCAU4 ≤ 1	← cluster = 2	13	2
ELSE IF:			
NP3RTCON ≤ 0 AND NP2WALK > 0 AND NP3PSTBL ≤ 2 AND NP2HWRT > 0 AND NP2SWAL > 1 AND SCAU6 ≤ 2	← cluster = 2	10	0
ELSE IF:			
NHY > 1 AND NP3KTRML ≤ 0 AND PN3RIGRL > 1 AND NP3RTARU = (0,1]	← cluster = 2	38	23
ELSE IF:			
NP3RTCON ≤ 0 AND NP3HMOVL ≤ 0 AND NP3RTARU ≤ 0 AND NP3TTAPR > 0	← cluster = 2	23	0
ELSE IF:			
NP3PSTBL > 2 AND NP2DRES > 0 AND NP3RTARU > 1	← cluster = 2	7	0
ELSE IF:			
NP3RTCON = (0,1] AND NHY > 1 AND NP3FTAPR > 1 AND NP3RTARL ≤ 0 AND NP2HOBB ≤ 1 AND NP2SPCH > 0 AND NP3FACXP > 0 AND NP3RTALU ≤ 0	← cluster = 2	24	11
ELSE IF:			
NP3PRSPL ≤ 0 AND NP3RTCON ≤ 0 AND NP3POSTR ≤ 2 AND SCAU18 ≤ 1 AND NP3RIGLL ≤ 0 AND NP3RTARU ≤ 0 AND SCAU3 ≤ 0	← cluster = 2	20	4

References

Agrawal, R., Imieliński, T., Swami, A. (1993). Mining association rules between sets of items in large databases. In *ACM sigmod record* (Vol. 22, pp. 207–216). ACM.

Appice, A. (2017). Towards mining the organizational structure of a dynamic event scenario. *Journal of Intelligent Information Systems*, 1–29.

Appice, A., & Malerba, D. (2016). A co-training strategy for multiple view clustering in process mining. *IEEE Trans Services Computing*, *9*(6), 832–845.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J.M., Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, *46*(1), 243–256.

Bence, J.R. (1995). Analysis of short time series: correcting for autocorrelation. *Ecology*, *76*(2), 628–639.

Bezdek, J.C. (1981). *Pattern recognition with fuzzy objective function algorithms*. New York: Plenum Press.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on computational learning theory, ACM, New York, NY, USA, COLT'* (Vol. 98, pp. 92–100).

Broder, A.Z., Glassman, S.C., Manasse, M.S., Zweig, G. (1997). Syntactic clustering of the web. *Computer Networks and ISDN Systems*, *29*(8-13), 1157–1166.

Cai, X., Nie, F., Huang, H. (2013). Multi-view k-means clustering on big data. In *Proceedings of the 23rd international joint conference on artificial intelligence IJCAI 2013, Beijing, China, August 3-9, 2013* (pp. 2598–2604).

Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, *3*(1), 1–27.

Ceravolo, R., Rossi, C., Kiferle, L., Bonuccelli, U. (2010). Nonmotor symptoms in Parkinson's disease: the dark side of the moon. *Future Neurology*, *5*(6), 851–871.

Chaudhuri, K., Kakade, S.M., Livescu, K., Sridharan, K. (2009). Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th annual international conference on machine learning, ICML 2009* (pp. 129–136).

Choi, E., Schuetz, A., Stewart, W.F., Sun, J. (2017). Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, *24*(2), 361–370.

Cleuziou, G., Exbrayat, M., Martin, L., Sublemontier, J. (2009). CoFKM: a centralized method for multiple-view clustering. In *Proceedings of the ninth IEEE international conference on data mining (ICDM 2009), miami, florida, USA, 6-9 December 2009* (pp. 752–757).

Dalrymple-Alford, J., MacAskill, M., Nakas, C., Livingston, L., Graham, C., Crucian, G., Melzer, T., Kirwan, J., Keenan, R., Wells, S., et al. (2010). The moCA: well-suited screen for cognitive impairment in Parkinson disease. *Neurology*, *75*(19), 1717–1725.

Davies, D.L., & Bouldin, D.W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *1*(2), 224–227.

De Alba, E., Mendoza, M., et al. (2007). Bayesian forecasting methods for short time series. *The International Journal of Applied Forecasting*, *8*, 41–44.

De Vine, L., Zuccon, G., Koopman, B., Sitbon, L., Bruza, P. (2014). Medical semantic similarity with a neural language model. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management* (pp. 1819–1822). ACM.

Dorsey, E., Constantinescu, R., Thompson, J., Biglan, K., Holloway, R., Kieburtz, K., Marshall, F., Ravina, B., Schifitto, G., Siderowf, A., et al. (2007). Projected number of people with Parkinson disease in the most populous nations, 2005 through 2030. *Neurology*, *68*(5), 384–386.

Ernst, J., & Bar-Joseph, Z. (2006). STEM: a tool for the analysis of short time series gene expression data. *BMC Bioinformatics*, *7*(1), 191.

Ernst, J., Nau, G.J., Bar-Joseph, Z. (2005). Clustering short time series gene expression data. *Bioinformatics*, *21*(suppl 1), i159–i168.

Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD* (Vol. 96, pp. 226–231).

European Parkinson's Disease Association (2016). http://www.epda.eu.com/, accessed: 2016/07/01.

Foltynie, T., Brayne, C., Barker, R.A. (2002). The heterogeneity of idiopathic Parkinson's disease. *Journal of Neurology*, *249*(2), 138–145.

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

Gatsios, D., Rigas, G., Miljkovic, D., Seljak, B.K., Bohanec, M. (2016). m-health platform for Parkinson's disease management. In *Proceedings of 18th international conference on biomedicine and health informatics CBHI*.

Gil, D., & Johnson, M. (2009). Diagnosing Parkinson by using artificial neural networks and support vector machines. *Global Journal of Computer Science and Technology*, *9*(4), 63–71.

Goetz, C.G., Tilley, B.C., Shaftman, S.R., Stebbins, G.T., Fahn, S., Martinez-Martin, P., Poewe, W., Sampaio, C., Stern, M.B., Dodel, R., et al. (2008). Movement Disorder Society-sponsored Revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS): scale presentation and clinimetric testing results. *Movement Disorders*, *23*(15), 2129–2170.

Goetz, C.G., Luo, S., Wang, L., Tilley, B.C., LaPelle, N.R., Stebbins, G.T. (2015). Handling missing values in the MDS-UPDRS. *Movement Disorders*, *30*(12), 1632–1638.

Greene, D., Doyle, D., Cunningham, P. (2010). Tracking the evolution of communities in dynamic social networks. In *Advances in social networks analysis and mining (ASONAM), 2010* (pp. 176–183). IEEE.

Guthrie, D., Allison, B., Liu, W., Guthrie, L., Wilks, Y. (2006). A closer look at skip-gram modelling. In *Proceedings of the 5th international conference on language resources and evaluation (LREC-2006)* (pp. 1–4).

He, X., Kan, M.Y., Xie, P., Chen, X. (2014). Comment-based multi-view clustering of Web 2.0 items. In *Proceedings of the 23rd international conference on World Wide Web* (pp. 771–782): ACM.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*(1), 193–218.

Hughes, A.J., Daniel, S.E., Kilford, L., Lees, A.J. (1992). Accuracy of clinical diagnosis of idiopathic Parkinson's disease: a clinico-pathological study of 100 cases. *Journal of Neurology Neurosurgery & Psychiatry*, *55*(3), 181–184.

Imhoff, M., Bauer, M., Gather, U., Löhlein, D. (1998). *Time series analysis in intensive care medicine*. Tech. rep. SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund.

Kaufman, L., & Rousseeuw, P. (1987). *Clustering by means of medoids*. North-Holland.

Kaufman, L., & Rousseeuw, P.J. (1990). *Finding groups in data. an introduction to cluster analysis*. Wiley Series in Probability and Mathematical Statistics Applied Probability and Statistics.

Kumar, A., & III, H.D. (2011). A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th international conference on machine learning, ICML* (pp. 393–400).

Lewis, S., Foltynie, T., Blackwell, A., Robbins, T., Owen, A., Barker, R. (2005). Heterogeneity of Parkinson's disease in the early clinical stages using a data driven approach. *Journal of Neurology, Neurosurgery & Psychiatry*, *76*(3), 343–348.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H. (2016). Feature selection: a data perspective. arXiv:160107996.

Lin, J., Keogh, E., Wei, L., Lonardi, S. (2007). Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, *15*(2), 107–144.

Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J. (2010). Understanding of internal clustering validation measures. In *Proceedings of IEEE 10th international conference on data mining (ICDM)* (pp. 911–916).

Liu, Y., Li, W., Tan, C., Liu, X., Wang, X., Gui, Y., Qin, L., Deng, F., Hu, C., Chen, L. (2014). Meta-analysis comparing deep brain stimulation of the globus pallidus and subthalamic nucleus to treat advanced Parkinson disease: a review. *Journal of Neurosurgery*, *121*(3), 709–718.

Ma, L.Y., Chan, P., Gu, Z.Q., Li, F.F., Feng, T. (2015). Heterogeneity among patients with Parkinson's disease: cluster analysis and genetic association. *Journal of the Neurological Sciences*, *351*(1), 41–45.

Marek, K., Jennings, D., Lasch, S., Siderowf, A., Tanner, C., Simuni, T., Coffey, C., Kieburtz, K., Flagg, E., Chowdhury, S., et al. (2011). The Parkinson's Progression Markers Initiative (PPMI). *Progress in Neurobiology*, *95*(4), 629–635.

Michalski, R.S. (1983). A theory and methodology of inductive learning. *Artificial Intelligence*, *20*(2), 111–161.

Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv:13013781.

Minarro-Giménez, J.A., Marín-alonso, O., Samwald, M. (2013). Exploring the application of deep learning techniques on medical text corpora. *Studies in Health Technology and Informatics*, *205*, 584–588.

Murugesan, S., Bouchard, K., Chang, E., Dougherty, M., Hamann, B., Weber, G.H. (2017). Multi-scale visual analysis of time-varying electrocorticography data via clustering of brain regions. *BMC Bioinformatics*, *18*(6), 236.

National Collaborating Centre for Chronic Conditions. (2006). *Parkinson's disease: national clinical guideline for diagnosis and management in primary and secondary care*. London: Royal College of Physicians.

Patel, S., Lorincz, K., Hughes, R., Huggins, N., Growdon, J., Standaert, D., Akay, M., Dy, J., Welsh, M., Bonato, P. (2009). Monitoring motor fluctuations in patients with Parkinson's disease using wearable sensors. *IEEE Transactions on Information Technology in Biomedicine*, *13*(6), 864–873.

PD_manager: m-Health platform for Parkinson's disease management (2015). EU Framework Programme for Research and Innovation Horizon 2020, Grant number 643706, 2015–2017. http://www.parkinson-manager.eu/.

Ramani, R.G., & Sivagami, G. (2011). Parkinson disease classification using data mining algorithms. *International Journal of Computer Applications*, *32*(9), 17–22.

Reijnders, J., Ehrt, U., Lousberg, R., Aarsland, D., Leentjens, A. (2009). The association between motor subtypes and psychopathology in Parkinson's disease. *Parkinsonism & Related Disorders*, *15*(5), 379–382.

Riviere, C.N., Reich, S.G., Thakor, N.V. (1997). Adaptive Fourier modeling for quantification of tremor. *Journal of Neuroscience Methods*, *74*(1), 77–87.

Rousseeuw, P.J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.

Samà, A., Pérez-López, C., Rodríguez-martín, D., Moreno-aróstegui, J.M., Rovira, J., Ahlrichs, C., Castro, R., Graća, R., Guimarães, V., et al. (2015). A double closed loop to enhance the quality of life of Parkinson's disease patients: REMPARK system. *Innovation in Medicine and Healthcare*, *2014*(207), 115.

Schieb, L.J., Mobley, L.R., George, M., Casper, M. (2013). Tracking stroke hospitalization clusters over time and associations with county-level socioeconomic and healthcare characteristics. *Stroke*, *44*(1), 146–152.

SENSE-PARK (2016). Project's website: http://www.sense-park.eu/, accessed: 2016/07/01.

Stecher, J., Janssen, F., Fürnkranz, J. (2014). Separating rule refinement and rule selection heuristics in inductive rule learning. In *Proceedings of machine learning and knowledge discovery in databases - European conference, ECML PKDD 2014* (pp. 114–129).

Szymański, A., Kubis, A., Przybyszewski, A.W. (2015). Data mining and neural network simulations can help to improve deep brain stimulation effects in Parkinson's disease. *Computer Science*, *16*(2), 199.

Timmer, J., Gantert, C., Deuschl, G., Honerkamp, J. (1993). Characteristics of hand tremor time series. *Biological Cybernetics*, *70*(1), 75–80.

Tzallas, A.T., Tsipouras, M.G., Rigas, G., Tsalikakis, D.G., Karvounis, E.C., Chondrogiorgi, M., Psomadellis, F., Cancela, J., Pastorino, M., Waldmeyer, M.T.A., et al. (2014). PERFORM: a system for monitoring, assessment and management of patients with Parkinson's disease. *Sensors*, *14*(11), 21,329–21,357.

Tzortzis, G., & Likas, A. (2009). Convex mixture models for multi-view clustering. In *Proceedings of the 19th international conference artificial neural networks - ICANN* (Vol. 2009, pp. 205–214).

Valmarska, A., Robnik-Šikonja, M., Lavrač, N. (2015). Inverted heuristics in subgroup discovery. In *Proceedings of the 18th international multiconference information society*.

Valmarska, A., Miljkovic, D., Lavrač, N., Robnik-Šikonja, M. (2016). Towards multi-view approach to Parkinson's disease quality of life data analysis. In *Proceedings of the 5th international workshop on new frontiers in mining complex patterns at ECML-PKDD2016*.

Valmarska, A., Lavrač, N., Fürnkranz, J., Robnik-Šikonja, M. (2017). Refinement and selection heuristics in subgroup discovery and classification rule learning. *Expert Systems with Applications*, *81*, 147–162.

Visser, M., Marinus, J., Stiggelbout, A.M., Van Hilten, J.J. (2004). Assessment of autonomic dysfunction in Parkinson's disease: the SCOPA-AUT. *Movement Disorders*, *19*(11), 1306–1312.

Washburn, R.A., Smith, K.W., Jette, A.M., Janney, C.A. (1993). The physical activity scale for the elderly (PASE): development and evaluation. *Journal of Clinical Epidemiology*, *46*(2), 153–162.

Xu, C., Tao, D., Xu, C. (2013). A survey on multi-view learning. *Neural Computing and Applications*, *23*(7–8), 2031–2038.

Zhao, J., Papapetrou, P., Asker, L., Boström, H. (2017). Learning from heterogeneous temporal data in electronic health records. *Journal of Biomedical Informatics*, *65*, 105–119.

Zhao, Z., & Liu, H. (2007). Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th international conference on machine learning* (pp. 1151–1157). ACM.

Chapter 6

Detection of Medications Change Patterns

This chapter extends the work presented in Chapter 5 by introducing a Relief-like algorithm for determining the symptoms that influence the change of Parkinson's disease patients' quality of life. The algorithm takes into account the temporal nature of symptoms data and detects influential symptoms that improve or worsen as the status of patients improves or degrades. We present a novel methodology for analysis of medication changes in Parkinson's disease progression data using multitask learning—where multiple related tasks are learned simultaneously on a shared attribute space—to simultaneously predict dosage changes of different groups of antiparkinson medications. Our aim is to establish a basis for more personalized medications therapy modifications based on the patients' symptoms. The chapter is divided into two sections: problem description and the published *Artificial Intelligence in Medicine* journal paper which addresses the described problem.

6.1 Problem Description

In Chapter 5 we addressed the issue of determining patterns of Parkinson's disease progression as well as patterns of medications changes that occur when the overall status of Parkinson's disease patients improves or degrades. However, we did not address exactly which symptoms changes cause the change in the patients' overall status and how the clinicians react with therapy modifications.

The determination of influential symptoms may help the clinicians to focus on a small set of most important symptoms, of which medications treatment would lead to a more stable status of the patient. In this chapter, we present our algorithm for determining the symptoms that affect the progression of the disease. The algorithm builds upon the work presented in Chapter 5 and is closely related to the feature evaluation algorithms Relief (Kira & Rendell, 1992; Kononenko, 1994; Robnik-Šikonja & Kononenko, 2003). We determine the importance of attributes based on the difference of their values when the overall status of the patient changes, i.e. when in two consecutive visits the patient is assigned to different clusters.

Similarly to our approach to determining the importance of symptoms for the overall disease progression, the feature evaluation algorithms Relief and ReliefF also compare feature values of similar instances from the same class and similar values from a different class. However, if applied to our problem, these algorithms cannot take into account the temporal progress of patients, i.e. they cannot track individual patients on their consecutive visits to the clinician. In effect, they show which attributes influence the initial assignment of patients into clusters, but reveal no information about the attributes which are the

most influential for changes of the patients' overall status (i.e. for the crossing of clusters of patients with a similar status).

Modifications of patients' therapies are motivated by the status of patients. The status is a result of the symptoms that develop due to the natural progression of the disease as well as the side-effect symptoms caused by the prolonged usage of antiparkinson medications. The goal of clinicians is to keep the patient's symptoms stable, preclude the side-effects, and prolong the patients' independence. Customarily, a clinician will lower the dosage of one group of medications and simultaneously increase the dosage of another group of medications. For example, clinicians usually start the treatment of motor symptoms in younger Parkinson's disease patients by introducing dopamine agonists. However, a known side effect of prolonged usage of dopamine agonists is impulsivity. When confronted with such conditions, the clinician will lower the dosage of dopamine agonists and simultaneously introduce levodopa in order to keep the motor symptoms stable. Clinicians' decisions for therapy modifications are made based on the patients' status and follow the official guidelines for the treatment of Parkinson's disease, e.g. (Ferreira et al., 2013; Fox et al., 2011; National Collaborating Centre for Chronic Conditions, 2006; Olanow, Watts, & Koller, 2001; Seppi et al., 2011).

These therapy modifications happen simultaneously and are a motivation for multitask learning. Compared to single-task learning, multitask learning can improve model generalization and prevent overfitting (Caruana, 1997). In this chapter, we address the issue of changes in antiparkinson medications (*yes/no* model) by modeling them with predictive clustering trees (PCT). The comprehensive models generated by PCT serve three purposes: i) identification of therapy modification scenarios, ii) identification of symptoms influencing the therapy modifications, and iii) determining subgroups of patients with shared similarities in their symptoms and therapy modifications. Multitask learning allows for determining what dosage changes of antiparkinson medications occurred simultaneously and by following the model we are able to determine the symptoms that affected these changes.

6.2 Related Publication

The rest of this chapter presents the *Artificial Intelligence in Medicine* journal paper.

Publication related to this contribution

Journal Paper

Valmarska, A., Miljkovic, D., Konitsiotis, S., Gatsios, D., Lavrač, N., & Robnik-Šikonja, M. (2018). Symptoms and medications change patterns for Parkinson's disease patients stratification. *Artificial Intelligence in Medicine (accepted).*

This publication contains the following contributions:

- We present an algorithm for determining the symptoms that are influential for the progression of the disease.
- We present a list of symptoms influencing the improvement or the degradation of the patients' status. This list is supported by a medical interpretation from our consulting clinician and references in the medical literature.
- We present a multitask learning based methodology for determining patterns of medication dosage changes based on the status of the patients. The methodology uses predictive clustering trees (PCTs) to model changes in patients' therapies.

- The presented methodology can be interpreted three-fold: i) the tree leaves present changes of medications therapies, ii) paths from the tree root to its leaves outline the symptoms influencing the changes of therapies, and iii) patients covered by the rules from the root of the tree to its leaves form groups of patients that are similar based on both their symptoms and their medications therapies.
- We empirically show that the changes of medications therapies can be modeled by multitask learning models. Predictive clustering trees are a good approach as in addition to the increased classification accuracy of the multitask model over the predictive performance of single task models, they also offer a simultaneous insight into what happens to the dosage of all antiparkinson medications.

The authors' contributions are as follows. The methodology was designed and developed by Anita Valmarska with the insights from Marko Robnik-Šikonja. Nada Lavrač has suggested using PCTs to model modifications of the patients' therapies. Anita Valmarska implemented the methodology and performed the experimental work. Marko Robnik-Šikonja and Nada Lavrač supervised the implementation of the algorithms. Dragana Miljkovic provided insights into data mining of Parkinson's disease data. Dimitris Gatsios and Spiros Konitsiotis helped with the interpretation of the results and their medical justification. All authors contributed to the text of the manuscript.

Symptoms and medications change patterns for Parkinson's disease patients stratification

Anita Valmarska(✉)[1,2], Dragana Miljkovic[1], Spiros Konitsiotis[3], Dimitris Gatsios[4], Nada Lavrač[1,2], and Marko Robnik-Šikonja[5]

[1] Jožef Stefan Institute, Jamova 39, Ljubljana, Slovenia
[2] Jožef Stefan International Postgraduate School, Jamova 39, Ljubljana, Slovenia
[3] University of Ioannina, Medical School, Department of Neurology, Ioannina, Greece
[4] University of Ioannina, Department of Biomedical Research, Ioannina, Greece
[5] University of Ljubljana, Faculty of Computer and Information Science, Slovenia
{anita.valmarska,dragana.miljkovic,nada.lavrac}@ijs.si
skonitso@uoi.gr, dgatsios@cc.uoi.gr, marko.robnik@fri.uni-lj.si

Abstract. Quality of life of patients with Parkinson's disease degrades significantly with disease progression. This paper presents a step towards personalized management of Parkinson's disease patients, based on discovering groups of similar patients. Similarity is based on patients' medical conditions and changes in the prescribed therapy when the medical conditions change. We present two novel approaches. The first algorithm discovers symptoms' impact on Parkinson's disease progression. Experiments on the Parkinson Progression Markers Initiative (PPMI) data reveal a subset of symptoms influencing disease progression which are already established in Parkinson's disease literature, as well as symptoms that are considered only recently as possible indicators of disease progression by clinicians. The second novelty is a methodology for detecting patterns of medications dosage changes based on the patient status. The methodology combines multitask learning using predictive clustering trees and short time series analysis to better understand when a change in medications is required. The experiments on PPMI data demonstrate that, using the proposed methodology, we can identify some clinically confirmed patients' symptoms suggesting medications change. In terms of predictive performance, our multitask predictive clustering tree approach is comparable to the random forest multitask model, but has an advantage of model interpretability.

Keywords: Parkinson's disease; analysis of disease progression; multitask learning; analysis of medications treatment; symptoms impact

1 Introduction

Data mining algorithms have been successfully used to learn predictive models and to discover insightful patterns in the data. Predictive and descriptive data mining approaches have been successfully used also in medical data analysis.

The use of data mining methods may improve diagnostics, disease treatment and detection of causes of diseases. In personalized healthcare [16], data mining can be used to improve drug recommendations and medical decision support, leading to reduced costs of medical treatment. The discovered patterns can provide the clinicians with new insights regarding the status of the treated patients and can support decisions regarding therapy recommendations.

Parkinson's disease is the second most common neurodegenerative disease (after Alzheimer's disease) that affects many people worldwide. Due to the death of nigral neurons, patients experience both motor and non-motor symptoms, affecting their quality of life. The reasons for the cell death are still poorly understood, and there is currently no cure for Parkinson's disease. Physicians try to manage patients' symptoms by introducing medications therapies, using antiparkinson medications. Physicians need to carefully prescribe medications therapies since the prolonged intake—in particular of higher dosages of antiparkinson medications—can have significant side-effects.

Changes of the status of Parkinson's disease patients through time is a result of the natural progression of the disease and the medications that the patients are prescribed in order to keep their status stable as long as possible. Physicians follow the guidelines for therapy prescription and the response of patients to medications is usually recorded in clinical studies using simple statistical methods. For example, in our previous work [43], we describe the disease progression of a patient who starts with a good status and only of one type of medications (MAO-B inhibitors). As the disease progressed and the patients motor symptoms worsened, the clinician started the treatment with another type of medications (dopamine agonists) and was successful in keeping the motor symptoms as tremor, bradykinesia, and rigidity stable for about two years. As the effectiveness of these medications wore off, the clinician was forced into introducing the third group of medications (levodopa).

To the best of our knowledge, data mining techniques have not yet been used for analyzing clinicians' decisions of changing drug prescription as a reaction to the change of patients' symptoms when using antiparkinson medications through prolonged periods of time. A possible reason for little data mining research in the field of Parkinson's disease progression may be the unavailability of a monotone measure/test that determines the stages of Parkinson's disease, as the currently used Hoehn and Yahr scale [15] determines the stages of Parkinson's disease through a subjective evaluation of clinicians and response of patients to the prescribed medications. This paper uses multitask learning with predictive clustering trees [4] on short time series data—describing the patients' status at multiple time points—in order to determine the symptoms that trigger the physicians' decisions to modify the medications therapy. We consider trigger symptoms to be the symptoms that a patient cannot tolerate and the physician is pressed to change the medications therapy in order to control them. The proposed methodology addresses the task of determining subgroups of patients with similar symptoms and therapy. As each patient usually receives drugs from

several different groups of medications, predicting their changes with multitask learning can lead to improved control over drug interactions.

This work significantly extends the conference paper [42] by extending the experiments, results, and their medical interpretation. We introduce a novel algorithm for determining the symptoms that have the highest influence on the change of the patients' status, which extends the methodology used to determine the status of Parkinson's disease patients based on an extensive set of symptoms [43, 44]. We present a solution to the problem of feature ranking with the aim of finding the most influential symptoms affecting the changed status of patients, which may help the clinicians to focus on a small set of the most important symptoms, whose medications treatment would lead to a more stable status of the patient. Our research provides references to the already known findings in Parkinson's disease literature, as well as references to findings about possible influential symptoms that have only recently started being discussed in the Parkinson's disease medical community as early indicators of Parkinson's disease progression. We significantly extend the experiments with PCT models, analyze different sets of attributes, and discuss reasons for particular medications dosage change patterns from the medical perspective. The consulting clinician takes into account trigger symptoms from the trees as well as the patients' overall status concerning their motor and non-motor symptoms.

This paper is structured into six sections. After presenting the background and related work in Section 2, Section 3 describes the Parkinson's Progression Markers Initiative (PPMI) symptoms data set [24], together with the data describing the medications used for symptoms control, available from the so-called PPMI concomitant medications log data set. Section 4 outlines our methodology. In Section 4.1 we present a new algorithm for determining the most influential symptoms. Section 4.2 proposes a methodology for analyzing Parkinson's disease symptoms by learning predictive clustering trees from short data sequences. Results are presented in Section 5. Section 5.1 presents the most influential symptoms, while Section 5.2 describes the results of applying the proposed methodology to the detection of changes in symptoms-based clustering of patients, connected to the changes in medications therapies and finding patterns of symptoms which trigger therapy modifications. In Section 5.3 we explore the influence of the above-mentioned symptoms on clinicians' decisions regarding the modification of dosages of prescribed medications. Finally, Section 6 presents the conclusions and plans for further work.

2 Background and Related Work

Our work is related to several subareas of data analysis. We first present approaches to Parkinson's disease data analysis in Section 2.1 and Parkinson's disease progression in Section 2.2. In Parkinson's disease management, several groups of medications are used together. We apply multitarget modeling with predictive clustering trees to capture their joint effects and discuss related work from this area in Section 2.3. We are interested in the importance of symptoms

affecting the overall status of the disease, which is a problem addressed in feature ranking/evaluation research. We compare and contrast the algorithm we propose with existing approaches in Section 2.4.

2.1 Parkinson's disease data analysis

Data mining research in the field of Parkinson's disease can be divided into three groups: classification of Parkinson's disease patients, detection of Parkinson's disease symptoms (computational assessment from e.g., wearable sensors), and detection of subtypes of Parkinson's disease patients, as discussed below.

Due to the overlap of Parkinson's disease symptoms with other diseases, only 75% of clinical diagnoses of Parkinson's disease are confirmed to be idiopathic Parkinson disease at autopsy [17]. Classification techniques offer decision support to specialists by increasing the accuracy and reliability of diagnosis and reducing possible errors. Gil and Johnson [13] use Artificial Neural Networks (ANN) and Support Vector Machines (SVM) to distinguish Parkinson's disease patients from healthy subjects. Ramani and Sivagami [29] compare the effectiveness of different data mining algorithms in the diagnosis of Parkinson's disease patients, where the data set consists of 31 people, 23 of which are Parkinson's disease patients.

Tremor is a symptom strongly associated with Parkinson's disease. Several approaches to computational assessment of tremor have been proposed. Methods such as time series analysis [41], spectral analysis [33], and non-linear analysis [33] have addressed tremor detection and quantification. Many recent works are based on body fixed sensors (BFS) for long-term monitoring of patients [26].

Parkinson's disease is a heterogeneous neurodegenerative condition with different clinical phenotypes, genetics, pathology, brain imaging characteristics and disease duration [11]. This variability indicates the existence of disease subtypes. Using k-means clustering, Ma et al. [23] identify four groups of Parkinson's disease patients which is consistent with the conclusions from [22, 31]. This division of Parkinson's patients into homogeneous subgroups was done on symptoms data recorded only once for each patient. It does not take into account the progression of the disease and changes in the patients' status due to the medications treatment. Our analysis uses a different data set (see Section 3) which allows us to take these issues into account.

Classification and clustering models usually focus on diagnosing new patients. None of the listed methods follow the progression of the disease, and to the best of our knowledge, no data mining research in the field of Parkinson's disease analyzed the development of the disease in combination with the medications that the patients receive. Identification of groups of patients based on how they react to a certain therapy can be helpful in the assignment of personalized therapies and more adequate patient treatment. To this end, we propose a methodology for determining trigger symptoms, which influence the physician's decision about therapy modification. In addition, our methodology aims to uncover the side-effects of the modified therapy.

2.2 Parkinson's disease progression

There are no specific medical tests to determine the progression of Parkinson's disease for an individual patient. Currently, the clinicians commonly use the Hoehn and Yahr scale system [15] to describe the progression of Parkinson's disease symptoms. This evaluation can be seen as the clinicians' aggregate evaluation of the patient's motor status. Patient's status changes through time and even though the status of the patient is going to get worse during their treatment, there are periods where carefully prescribed medications therapies can cause an improvement of the patient's overall status. This improvement can be reflected in both the patient's *motor* and *non-motor* symptoms.

In our earlier work [43, 44], we first used unsupervised learning (k-means clustering) to divide Parkinson's disease patients from the PPMI study into three groups with similar severity of their *motor* and *non-motor symptoms*. We then applied supervised classification rule learning techniques to obtain descriptions for each of the obtained groups. The results suggested that these groups can be described with the aggregated severity of their motor symptoms. In addition, the rules also contained the information about the status of their non-motor symptoms.

The three groups of patients were ordered according to the sum of evaluation values for their motor symptoms from MDS-UPDRS Part III (NP3SUM). The first cluster (*cluster 0*) consisted of patients whose motor symptoms were considered as normal, and the sum of MDS-UPDRS Part III was below 22. The second cluster (*cluster 1*) contained patients whose motor symptoms were slightly worse, and the sum of MDS-UPDRS Part III was between 22 and 42. In the third cluster (*cluster 2*) there were the patients whose sum of evaluation symptoms values from MDS-UPDRS Part III were higher than 42. Note that based on the sum of motor symptoms, the status of patients from *cluster 2* is worse than the status of patients from *cluster 0* and *cluster 1*.

The patients' symptoms are recorded regularly (on their visit to the clinicians) and based on these symptoms, at each visit, the patients are assigned to a cluster. Assignments to clusters may change during different visits to the clinicians. Following these assignments to clusters through their recorded visits to the clinicians gives an overview of the changes in the overall status and on the disease progresses through time.

The separation of patients into three groups provides the information about patients' status based on their aggregate score for the motor symptoms. Unfortunately, it does not provide any information about the symptoms that are particularly bothersome for the patients, and whose change would have the strongest impact on the assignment of patients into a given cluster.

The identification of symptoms that strongly influence the change of the patient's overall status (the patient's assignment to one of the clusters), can help clinicians to focus their attention to a smaller set of symptoms when deciding possible treatment modifications[1] of the patients. Using the real world data, our

[1] A treatment modification is any change in overall LEDD (levodopa equivalent daily dosage) (change of frequency intake, change of medications group etc.).

aim is to reveal the symptoms that are the most susceptible to improvement or decline when the overall status of the patient changes. When deciding on the modification of patient's treatment, the clinicians may consider these symptoms in order to keep patient's status stable as long as possible. We present the algorithm for identification of the most impactful symbols in Section 4.1.

2.3 Multitask learning

In multitask learning (MTL), multiple related tasks are learned simultaneously on a shared attribute space. Compared to single task learning, MLT can improve model generalization and prevent overfitting [6]. This is achieved by transfer of intermediate knowledge between jointly learned tasks, e.g., constructed relevant paths in tree-based models or important joint subconcepts in neural networks. In this way, the learning does not focus on a single task (thus preventing overfitting) and what is learned for one task can help other tasks (thus improving generalization).

Caruana et al. [7] use knowledge from the future to rank patients according to their risk to die from pneumonia. The shared attribute space consists of patients' at the time they are admitted to the hospital. The multiple tasks which are learned by the model are a set of hospital tests performed to determine whether the patients are of a risk of dying of pneumonia. Zhou et al. [51] use multitask learning to model Alzheimer's disease progression. They use two clinical/cognitive measures, Mini Mental State Examination (MMSE) and Alzheimer's Disease Assessment Scale cognitive subscale (ADAS-Cog) as multiple evaluations to determine the progression of the disease. Zhang et al. [50] propose a multitask model for prediction of multiple regression and classification variables in Alzheimer's disease, which takes advantage of the multimodal nature of patient's symptoms. Similarly to Parkinson's disease patients, Alzheimer's patients can be described by symptoms collected from multiple sources. All of these approaches use quantitative data "from the future" (values of tests taken in the future) to determine how the disease progresses. The authors take historical data and use multitask learning to predict the two years in the future results of two tests (the MMSE and the ADAS-Cog questionnaire). Using the baseline MRI, FDG-PET, and CSF data they estimate the disease progression by predicting these two values and predicting the conversion of patients with a mild cognitive disorder (MCI) to patients with Alzheimer's disease (AD). Unfortunately, there are no tests to appropriately measure the progression of Parkinson's disease. None of the above-mentioned methods look at the medications patients are receiving to decelerate the disease progression.

We use multitask learning with the aim to simultaneously predicting the values of several target attributes (medications in our case). We use a supervised learning method called predictive clustering trees (PCTs) [3, 4]. This method adapts the basic top-down induction of decision trees with clustering and allows for multitask learning. The PCT learning algorithm used is implemented in the CLUS data mining framework [4]. We obtain multitask decision trees, simultaneously predicting three target variables: change of levodopa dosage, change of

dopamine agonists dosage, and change of MAO-B inhibitors dosage, referring to three most important medication groups used in Parkinson's disease patient management. The PCT-based approach is described in Section 4.2, and evaluated in Sections 5.2 and 5.3.

2.4 Feature evaluation

Feature subset selection can improve the accuracy, efficiency, applicability, and comprehensibility of a learning process and its resulting model [2]. For this reason, many feature subset selection approaches have been proposed. In general, three types of feature selection methods exist: wrapper, filter, and embedded methods. Wrapper methods use the performance of a given learning algorithm as the criterion to include/exclude attributes. Embedded methods use feature selection as an integral part of their learning process. Filter methods introduce some external criterion independent of the predictor. They evaluate features according to that criterion, which allows for ranking of features and selection of a suitable subset. This is fit for our purpose.

Our approach to determining the importance of symptoms for the overall disease progression is strongly related to the well-known Relief family of algorithms [19, 34, 32]. These algorithms evaluate attributes based on their ability to distinguish between similar instances with different class values. Contrary to the majority of feature evaluation heuristics (e.g., information gain, gini index, etc.) that assume conditional independence of attributes w.r.t. the target variable, the Relief approaches do not make this assumption and are suitable for problems that involve feature interaction. The Relief algorithms randomly select an instance and find the nearest instance from the same class and nearest instances from different classes. When comparing feature values of near instances the algorithm rewards features that separate instances with different class values and punishes features that separate instances with the same class value. The whole process is repeated for large enough sample. The approach we propose also uses similar instances but uses cluster membership as a criterion for similarity instead of a distance in the feature space. When updating the importance of features our approach assesses joint transitions from one cluster to another or from better patient status to a worse one, while Relief algorithms use similarities in target variable.

Some recent feature selection approaches try to explore the interconnection between the features by exploring the similarity graph of features [30, 38]. Other approaches pose feature selection as an optimization problem, for example, Sun et al. [40] use optimization in combination with a game theory based method. Our approach also uses a graph of transitions between clusters to assess similarity of patients, but we work in an unsupervised scenario and use time order of patients' visits as links between nodes. Details are explained in Sections 4.1 and 4.2.

3 Parkinson's Disease Data Set

In this paper we use the PPMI data collection [24] gathered in the observational clinical study to verify progression markers in Parkinson's disease. In Section 3.1 we present the PPMI symptoms data sets and in Section 3.2 we present the medications data used in the experiments. As there are altogether 114 attributes in described data sets, in their everyday practice, physicians focus on a subset of chosen symptoms to follow the development of the disease and decide when to intervene with medication modifications. The symptoms which are in the focus of physician's attention are discussed in Section 3.3.

3.1 PPMI symptoms data sets

The medical condition and the quality of life of a patient suffering from Parkinson's disease is determined using the Movement Disorder Society (MDS)-sponsored revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS) [14]. This is a four-part questionnaire consisting of 65 questions concerning the development of the disease symptoms. Part I consists of questions about the "non-motor experiences of daily living". These questions address complex behaviors, such as hallucinations, depression, apathy, etc., and patient's experiences of daily living, such as sleeping problems, daytime sleepiness, urinary problems, etc. Part II expresses "motor experiences of daily living". This part of the questionnaire examines whether the patient experiences speech problems, the need for assistance with the daily routines such as eating or dressing, etc. Part III is referred to as the "motor examination", while Part IV concerns "motor complications", which are mostly developed when the main antiparkinson drug levodopa is used for a longer time period. Questions from the MDS-UPDRS represent symptoms characteristic for Parkinson's disease, while their answers indicate the symptom's severity that a patient is experiencing. Each answer is given on a five-point Likert scale, where 0 = normal (patient's condition is normal, the symptom is not present), and 4 = severe (symptom is present and severely affects the independent functioning of the patient).

Cognitive state of a patient is determined using the Montreal Cognitive Assessment (MoCA) [8] questionnaire consisting of 11 questions (maximum 30 points), assessing different cognitive domains. In addition to the MoCA data, physicians also use the Questionnaire for Impulsive-Compulsive Disorders (QUIP) [48] to address four major and three minor impulsive-compulsive disorders.

Scales for Outcomes in Parkinson's disease—Autonomic (SCOPA-AUT) is a specific scale to assess autonomic dysfunction in Parkinson's disease patients [45]. Physical Activity Scale for the Elderly (PASE) [46] is a questionnaire which is a practical and widely used approach for physical activity assessment in epidemiologic investigations. Cognitive Categorization (COGCAT) is a questionnaire filled in by clinicians evaluating the cognitive state and possible cognitive decline of patients. The above data sets are periodically updated to allow the clinicians to monitor patients' disease development through time. Answers to the questions from each questionnaire form the vectors of attribute values.

Table 1 presents a summary of the symptoms data sets used in our research.[2] It lists the number of considered questions from each questionnaire, the range of attribute values, and the nature of the attribute values. Answers to the questions from questionnaires presented in Table 1 represent the combined set of symptoms used in our research to determine the status of Parkinson's disease patients. The total number of symptoms from the mentioned questionnaires is 114.

Answers to the considered questions are ordered values and, with the exception of MoCA and PASE questions, larger values suggest higher symptom severity and decreased quality of life for Parkinson's disease patients.

Table 1. Characteristics of the questionnaire data used in the analysis.

Questionnaire	Number of questions	Answers value range	Ordered values	Higher value indicates higher symptom severity
MDS-UPDRS Part I	6	0–4	Yes	Yes
MDS-UPDRS Part Ip	7	0–4	Yes	Yes
MDS-UPDRS Part II	13	0–4	Yes	Yes
MDS-UPDRS Part III	35	0–4	Yes	Yes
MDS-UPDRS Part IV	6	0–4	Yes	Yes
MoCA	11	0–1	Yes	No
PASE	7	1–2	Yes	No
SCOPA-AUT	21	0–3	Yes	Yes
COGCAT	4	0–1	Yes	Yes
QUIP	4	0–1	Yes	Yes
Total	114			

3.2 PPMI concomitant medications log

The PPMI data collection offers information about all of the concomitant medications that the patients used during their involvement in the study. We concentrate on whether a patient receives a therapy with antiparkinson medications and which combination of antiparkinson medications she/he received between two consecutive time points when the MDS-UPDRS and MoCA tests were administered. The three main families of drugs used for treating motor symptoms are levodopa, dopamine agonists, and MAO-B inhibitors [25].

The medications therapy for Parkinson's disease patients is highly personalized. Patients take different medications with personalized plans of intake. In order to be able to compare different therapies, dosages of Parkinson's disease medications are translated into a common Levodopa Equivalent Daily Dosage (LEDD).

[2] We do not have permission to share the data. Access to data can be obtained on the PPMI website: `http://www.ppmi-info.org/access-data-specimens/download-data/`.

3.3 Experimental symptoms data selected by clinicians

In their everyday practice, physicians use a vector of chosen symptoms to follow the development of the disease and decide when to intervene with medication modifications. They focus their attention on both *motor* and *non-motor* aspects of patients' quality of life. Physicians evaluate the motor aspect of patient's quality of life using the following symptoms: *bradykinesia*, *tremor*, *gait*, *dyskinesia*, and *ON/OFF fluctuations*. The *non-motor* aspect of patient's quality of life is determined using *daytime sleepiness*, *impulsivity*, *depression*, *hallucinations*, and *cognitive disorder*. In addition to *motor* and *non-motor* symptoms, physicians also consider epidemiological symptoms which include *age*, *employment*, *living alone*, and *disease duration*. According to the collaborating clinicians, physicians are inclined to change the therapy of younger patients (younger than 65[3]), who are still active, who live alone, and for the patients diagnosed with Parkinson's disease for a shorter time (less than 8 years). For these patients, physicians will try more changes to the therapy in order to find the most suitable therapy, rather than therapy prolongation with increased medications dosage strategy which is applied to older Parkinson's disease patients.

In modifying the patient's medications based on the numerical evaluation of symptoms, the physicians decide whether the symptom is *problematic* and needs their immediate attention or not. Table 2 presents the *motor* and *non-motor* symptoms influencing the physicians' decisions for medications modifications, the data sets they are part of, and the intervals of values that are considered *normal* or *problematic* for Parkinson's disease patients. For example, the value of *tremor* is defined as the mean value of all questions concerning tremor from MDS-UPDRS Part II and Part III. Intervals of *normal* and *problematic* values are determined by the clinical expert. For all UPDRS items, value 0 is normal, value 1 is slight or minor, value 2 is mild, 3 is moderate and 4 is severe. Thus, in most cases and given the progressive nature of Parkinson's disease, values 0 and 1 of symptoms are not problematic and are baring for the patients, but become annoying and hampering when they progress in the range 2–4: this leads to distinguishing between values *normal* and *problematic* [14]. The selection of these 10 *motor* and *non-motor* symptoms, and *age* as an *epidemilogical* symptom, constituted the subset of attributes considered in the experiments presented in Section 5.2. The reason for excluding *employment*, *living alone*, and *cognitive disorder*, which could be important *epidemilogical* attributes, is that the PPMI data collection does not have data about patients' employment and living arrangements. We omitted the *cognitive disorder* attribute due to its values in the database, which were either *normal* or *missing*[4].

For each patient in the data set, the *motor* and *non-motor* symptoms data were obtained and updated periodically (on each patient's visit to the clinician's),

[3] Retirement age for men (`https://en.wikipedia.org/wiki/Retirement_age`).

[4] We explored the option of handling 'structurally missing' data, where the *cognitive disorder* attribute was kept in the final analysis. Across all attributes a new attribute value *missing* was introduced. The generated model had a lower classification accuracy than the model presented in Section 5.2.

Table 2. Description of *motor* (upper part) and *non-motor* (lower part) symptoms used by Parkinson's disease physicians in everyday practice to estimate patient's quality of live. The values intervals (*normal* and *problematic*) are defined by the clinician.

Symptom	Data set	Question number	Normal values interval	Problematic values interval
bradykinesia	MDS-UPDRS Part III	3.14	0–1	2–4
tremor	MDS-UPDRS Part II and III	mean value	0	1–4
gait	MDS-UPDRS Part III	3.10	0–1	2–4
dyskinesia	MDS-UPDRS Part IV	4.3	0–1	2–4
ON/OFF fluctuations	MDS-UPDRS Part IV	4.5	0	1–4
daytime sleepiness	MDS-UPDRS Part I	1.8	0–1	2–4
impulsivity	QUIP	SUM	0–1	≥ 2
depression	MDS-UPDRS Part I	1.3	0–1	2–4
hallucinations	MDS-UPDRS Part I	1.2	0–1	2–4
cognitive disorder	MoCA	SUM	26–30	<26

providing the clinicians with the opportunity to follow the development of the disease. The data set contains 897 instances, containing information about 368 PPMI patients. Most of the considered patients have records about two or three visits to the clinician. The maximum number of visits is 4.

4 Methodology

In this section, we present two methodologies: a methodology for patients' symptoms impact on the Parkinson's disease progression and a methodology for detecting medications dosage change patterns as a result of the patient's symptoms. Section 4.1 outlines an algorithm for determining which symptoms have the strongest impact on the patients' overall status. The patients' overall status is determined by the severity of a large set of symptoms (see Section 3.1). This methodology is closely related to our previous research on Parkinson's disease progression, shortly summarized in Section 2.2 as well as the work done on feature evaluation (Section 2.4). Results from this methodology, i.e. a list of symptoms that our algorithm finds to have the strongest impact on the change of patients' overall status are presented in Section 5.1.

Section 4.2 presents our methodology for detecting medications dosage change patterns as a result of the patient's symptoms. This methodology serves two aims: detecting patterns of medications dosage changes based on the patient's overall status as well as identifying clinically confirmed symptoms suggesting medications change. Our methodology is related to the work done on multitask learning (Section 2.3). Results from the evaluation of the methodology on the set of symptoms data selected by clinicians are presented in Section 5.2.

4.1 Symptoms' impact on Parkinson's disease progression

This section outlines a pseudo code of the algorithm which estimates the impact of symptoms on the change of patients' overall status—their change of clusters. The most important symptoms found by this algorithm are presented in Section 5.1.

The *getAttrChangeProbabilities* function, presented in Algorithm 1, is a supervised approach that estimates the probabilities that feature (symptom) values changed when the patients' overall status also changed (i.e. when the patients have crossed clusters) or stayed the same (the patients have not changed clusters between two consecutive visits).

As the input Algorithm 1 takes **F**, patients' symptoms data described in Section 3.1, the index data set **I**, and the assigned cluster labels **c**. The patients' symptoms data **F** contains the information about the patients' symptoms values at different visits to the clinicians. It is a matrix of dimension n (number of instances) times $|\mathbf{A}|$ (number of considered symptoms). The features data set **F** contains the information on 114 *motor* and *non-motor* symptoms of Parkinson's disease patients. **F** rows represent the instances (patient p_i on visit v_{ij}), and the columns present patients' symptoms. The index data set **I** holds the instance indexes represented as a combination of patients and their visits. Vector **c** holds the information about the cluster to which a patient in a certain visit has been assigned to (i.e. c_{ij} marks the cluster patient p_i was assigned to on visit v_{ij}, see Section 2.2).

The output of the algorithm is two matrices, $attrChangeProbability$ and $attrSameProbability$, of dimension $K \times K \times |A|$ (K is the number of clusters), which hold the probabilities that an attribute will change value or stay the same for a certain cluster crossing, respectively.

The algorithm first initializes its working spaces and storage matrixes (lines 2 - 6). For each patient and for each two consecutive visits, the algorithm compares the assigned cluster labels for each instance (for each combination of (p_i, v_{ij}, c_{ij}) and $(p_i, v_{ij+1}, c_{ij+1})$) in lines 8 - 24. For each cluster change combination, the algorithm also takes note of what happens to the symptoms' values—whether they changed or stayed the same (lines 15 - 21). The recorded changes of symptoms and clusters are normalized with the total number of cluster crossing (lines 25 - 37) and the resulting probabilities are returned (line 38).

As a result of Algorithm 1, we get probabilities which reflect the impact of the attributes on cluster changes. This can serve in inference on the disease progression but also to select only the most influential attributes and thereby decrease the dimensionality of attribute space. We discuss the use of Algorithm 1 in Section 5.1.

4.2 Medications dosage change patterns

Our goal is to support physicians in their decisions regarding the patients' therapies. The physicians have several groups of medications at their disposal with which they try to preserve the good quality of patient's life. They use and switch between different groups of drugs and their dosages to treat different symptoms (e.g., levodopa is used for *motor* symptoms), and also to prevent overuse of any specific drug in order to reduce side-effects and undesired drug interactions. Our multitask learning approach based on Predictive Clustering Trees (PCTs) [4] (introduced in Section 2.3) allows for modeling of all medication groups simultaneously. By simultaneously predicting several target variables, the model

```
getAttrChangeProbabilities(F,I,c):
  Input      : F – concatenated view (feature data set);
               A – attribute space;
               I – indices of patient-visit combinations;
               c – assigned cluster labels;
  Parameters : K – number of clusters in c;
  Output     : attrChangeProbability;
               attrSameProbability;

  // Count for each cluster crossing. Matrix noOfCrossings^{K×K} is initialized to 0.
  noOfCrossings = {1:K, 1:K} ← 0
  // Number of value changes for each attribute and each cluster crossing.
  // Matrix attrChangeNo^{K×K×|A|} is initialized to 0.
  attrChangeNo = {1:K, 1:K, 1:|A|} ← 0
  // Probability of value changes for each attribute and each cluster crossing.
  // Matrix attrChangeProbability^{K×K×|A|} is initialized to 0.
  attrChangeProbability = {1:K, 1:K, 1:|A|} ← 0
  // Number of unchanged values for each attribute and each cluster crossing.
  // Matrix attrSameNo^{K×K×|A|} is initialized to 0.
  attrSameNo = {1:K, 1:K, 1:|A|} ← 0
  // Pobability of unchanged values for each attribute and each cluster crossing.
  // Matrix attrSameProbability^{K×K×|A|} is initialized to 0.
  attrSameProbability = {1:K, 1:K, 1:|A|} ← 0

  for patient in [1:p] do
      // consecutive visits for a given patient
      patientsVisits ← I[patient,1:allVisits[patient]]
      for v_j, v_{j+1} in patientsVisits do
          prevCluster ← c[patient][v_j]
          currCluster ← c[patient][v_{j+1}]
          incrementByOne(noOfCrossings[prevCluster, currCluster])
          prevFeatures ← F[v_j]
          currFeatures ← F[v_{j+1}]
          for attr in A do
              if differs(prevFeatures[attr],currFeatures[attr]) then
                  incrementByOne(attrChangeNo[prevCluster, currCluster][attr])
              end
              else
                  incrementByOne(attrSameNo[prevCluster, currCluster][attr])
              end
          end
      end
  end

  // Determine the probability of changed/unchanged attribute values
  // for each cluster crossing.
  for c1 in [1:K] do
      for c2 in [1:K] do
          clusterCrosses = noOfCrossings[c1,c2]
          for attr in attrChangeNo[c1,c2] do
              attrChanges = attrChangeNo[c1,c2][attr]
              attrChangeProbability[c1,c2][attr] = attrChanges / clusterCrosses
          end
          for attr in attrSameNo[c1,c2] do
              attrSame = attrSameNo[c1,c2][attr]
              attrSameProbability[c1,c2][attr] = attrSame / clusterCrosses
          end
      end
  end
  return attrChangeProbability, attrSameProbability
```

Algorithm 1: Assessment of feature impact on cluster changes.

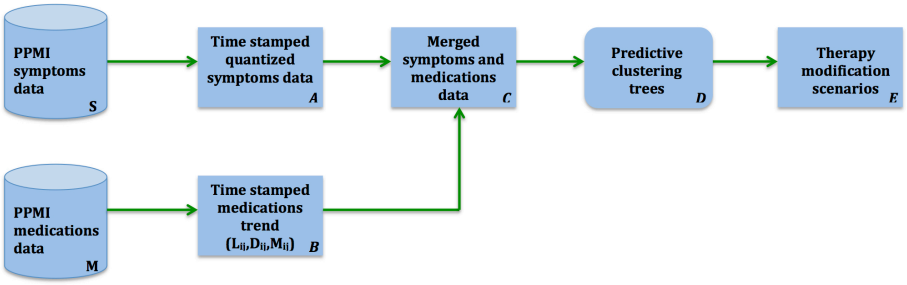


Fig. 1. Outline of the methodology for determining medications change patterns in PPMI data using predictive clustering trees.

allows physicians to observe the interactions between different groups of medications, which is not possible with univariate models. As training data, we use time-stamped symptoms and medications data. Figure 1 outlines the proposed five-step methodology, which uses symptoms data collected over time (i.e. over several patient's visits) and respective changes in medications therapies. Our goal is to identify symptoms scenarios for which the physicians need to consider modifications of therapies.

The input to the methodology are PPMI data sets of patient symptoms (described in Section 3.1) and the PPMI medications log data set (described in Section 3.2). The output of the methodology are patterns of patients' symptoms for which particular changes of medications were administered by the clinicians.

In step ***A*** we construct a time-stamped symptoms data set consisting of the symptoms (attributes) described in Section 3.1. This data set consists of patient-visit pairs (p_i, v_{ij}) describing the patients and their visits to the clinician.

In step ***B*** we construct a data set of medications changes which are represented with (p_i, m_{ij}, m_{ij+1}) tuples, where m_{ij} and m_{ij+1} are medication therapies of patient p_i in two consecutive visits, v_{ij} and v_{ij+1}. A patient receives a therapy which is any combination of levodopa, dopamine agonists, and MAO-B inhibitors. For each of the three medications groups, we determine whether its dosage in the time of visit v_{ij+1} has changed (*increased* or *decreased*) or remained unchanged with respect to the dosage at visit v_{ij}. The output of step ***B*** is a data set of medications changes, presented as tuples (L_{ij}, D_{ij}, M_{ij}), indicating whether between visits v_{ij} and v_{ij+1} a change of dosage in levodopa (L), dopamine agonist (D), or MAO-B inhibitors (M) took place.

In step **C** we concatenate the data sets obtained in steps ***A*** and ***B*** into a merged data set of symptoms and medications data. We use patient-visit pairs (p_i, v_{ij}) describing patient's symptoms at visit v_{ij} and the changes of medications in the same visit with respect to the next visit v_{ij+1}. These data consist of a set of attributes describing the condition of the patient, and three attributes (levodopa, dopamine agonists, and MAO-B) indicating the changes in their dosage, respectively. The set of symptoms describing the condition of the patient can

be preselected by clinicians, automatically selected, or a combination of both approaches.

The merged data set is used in step ***D*** to determine medications change patterns. The three medications groups are used as multitask variables (multiple classes) in the predictive clustering trees learning approach. We want to determine which symptoms influence decisions of physicians to modify the therapies that patients receive. The discovered therapy modifications patterns are analyzed by the physician in step ***E***.

Models produced by the PCT approach serve three aims: determining patterns of medications dosage changes, identification of Parkinson's disease symptoms suggesting medications dosage changes, and discovering groups of similar patients. These aims depend on the interpretation of the PCTs. Patterns of medications dosage changes are found in the leaves of the tree. Branches from the root of the tree to its leaves identify the symptoms influencing a particular pattern of medications dosage change, while patients experiencing these symptoms and medications dosage changes construct groups of patients that are similar based on both their symptoms and their medications therapy modifications.

We test the proposed methodology in two experimental settings, using two different symptoms data sets described in more detail below. In the first experimental setting, (Section 5.2) we use symptoms which were selected by our consulting clinician. In the second experimental setting (Section 5.3), we test the proposed methodology for determination of the symptoms' impact (see Section 4.1) and form a merged data set with symptoms selected by the clinician and the most influential symptoms according to Algorithm 1. We analyze symptom patterns for which the physicians modified the patients' therapies. We use the changes of the three medications groups as the target classification variables. Changes in dosage (increase or decrease) are marked with the class label *yes*, while unchanged drug dosages are marked with the class label *no*.

5 Evaluation

We split the evaluation of the proposed methodology into three parts. In Section 5.1 we use Algorithm 1 to find the most influential symptoms. In Section 5.2 we analyze the medications dosage change patterns detected from symptoms selected by clinicians (see Section 3.3). The most influential symptoms from Section 5.1 together with the symptoms selected by clinicians form a new data set and are analyzed in Section 5.3.

5.1 The most influential symptoms

When patients change clusters between two consecutive visits, this change can be considered as *positive* or *negative*. A *positive* cluster change occurs when between two consecutive visits a patient has crossed from a cluster with higher index (e.g., *cluster 2*) to a cluster with lower index (e.g., *cluster 1*). Given the cluster descriptions from [44], this change indicates that the overall status of

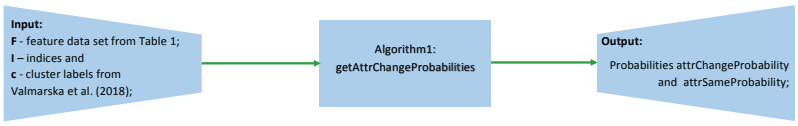


Fig. 2. A flowchart presenting the input, output, and method used for determining the most influential symptoms. Details about the input data can be found in Table 1 and [43, 44].

the patient concerning her/his motor symptoms has improved (indicated with lower MDS-UPDRS values). Contrarily, when in two consecutive visits a patient moves from a cluster with lower index (e.g., *cluster 1*) to a cluster with higher index (e.g., *cluster 2*) her/his overall status has worsen (as indicated by the sum of the motor symptom).

We ran the Algorithm 1 twice, the first time using numerical scores of symptoms (values 0-4 for MDS-UPDRS symptoms), and the second time using discretized values of the symptoms (*normal* and *problematic*). A flowchart presenting the input, output, and method used in this experimental setting is presented in Figure 2. For each run, the algorithm returned the probabilities of symptom changes and symptoms staying unchanged. Ranking the symptoms by the decreased probability of symptom changes and intersecting the top 25[5] features we get a list of symptoms that have the strongest impact on cluster changes. These are symptoms that have most frequently changed values, and whose change of values brought significant improvement (from *problematic* to *normal*) or decline (from *normal* to *problematic*).

Table 3 presents the intersection of lists obtained by two runs of the algorithm, for symptoms whose values have changed most frequently when a cluster change has occurred. The symptoms are presented with their code names from the PPMI data collection and with their descriptions. The results are ordered according to the decreased probability of cluster changes (weighted *positive* and *negative* changes).

We can note that the upper part of Table 3 is populated with the motor symptoms from MDS-UPDRS Part III. This is not surprising since as we mentioned above, the obtained clusters were ordered in accordance with the aggregate score of their motor symptoms from MDS-UPDRS Part III. In addition to the influential motor symptoms, the algorithm finds also a subset of influential non-motor symptoms whose values vary as the overall status of the patient's changes.

In practice, *positive* and *negative* changes are not treated equally and may not be caused by the same symptoms. Clinicians try to avoid *negative* changes and actively promote *positive* changes. We first report symptoms indicating positive changes, followed by the symptoms indicating negative changes.

[5] The number of top-ranked features was set experimentally so that the length of the intersection list is sufficiently informative and manageable for clinicians.

Table 3. List of most influential symptoms according to Algorithm 1. The symptoms are ordered according to their average rank of positive and negative impact.

PPMI attribute	Attribute description	PPMI data set	Attribute importance for cluster change
NP3BRADY	Bradykinesia	MDS-UPDRS Part III	0.314
NP3TTAPL	Toe tapping (left)	MDS-UPDRS Part III	0.297
NP3RTCON	Constancy of rest	MDS-UPDRS Part III	0.291
NP3FACXP	Facial expression	MDS-UPDRS Part III	0.282
NP3FTAPL	Finger tapping (left)	MDS-UPDRS Part III	0.273
NP3FTAPR	Finger tapping (right)	MDS-UPDRS Part III	0.255
NP3PRSPL	Hand pronation/supination (left)	MDS-UPDRS Part III	0.244
NP3TTAPR	Toe tapping (right)	MDS-UPDRS Part III	0.239
NP3PRSPR	Hand pronation/supination (right)	MDS-UPDRS Part III	0.203
NP1SLPN	Sleep problems (night)	MDS-UPDRS Part Ip	0.155
NP1SLPD	Daytime sleepiness	MDS-UPDRS Part Ip	0.147
NP2HWRT	Handwriting	MDS-UPDRS Part II	0.144
NP1FATG	Fatigue	MDS-UPDRS Part Ip	0.138
NP1URIN	Urinary problems	MDS-UPDRS Part Ip	0.134
NP1PAIN	Pain and other sensations	MDS-UPDRS Part Ip	0.117
MCATOT	MoCA total score (cognition)	MoCA	0.097

Figure 3 presents the symptoms whose values improve most frequently when the patients make a positive cluster change (their overall status between two consecutive visits improves). The results suggest that in over 37% of cases when the patient's status improves, also the value of their constancy of rest improves (NP3RTCON). The second most frequently improved symptom is bradykinesia (NP3BRADY), followed by the finger tapping in the left hand (NP3FTAPL).

Figure 4 presents the results for the symptoms whose values degrade most frequently when the patients make a negative cluster change and their overall status between two consecutive visits worsens. The results suggest that in over 30% of cases when the patients' status worsens, they experience problems with toe taping, facial expression, and bradykinesia.

Rigidity is a relevant and bothersome symptom for patients that was not detected by Algorithm 1. A reason for this omission may be the fact that rigidity is reported through five questions from MDS-UPDRS Part III (both hands, both legs, and neck). Patients can experience rigidity problems on different parts of the body and each of this parts may not be statistically strong enough to be ranked high by Algorithm 1. A way to alleviate this problem would be to combine answers from multiple questions concerning the same underlying symptom before running Algorithm 1. We plan such detailed analysis for our further work as well as a selection of two separate lists of symptoms which improve or decline most frequently. This could lead to distinguishing the symptoms for which change of medications dosage is the most effective, as well as those who are most inclined to worsening as the disease progresses.

Similarly to our approach to determining the importance of symptoms for the overall disease progression, the feature evaluation algorithms Relief and ReliefF [19, 34] also compare feature values of similar instances from the same class and similar values from a different class. Relief and ReliefF reward the features

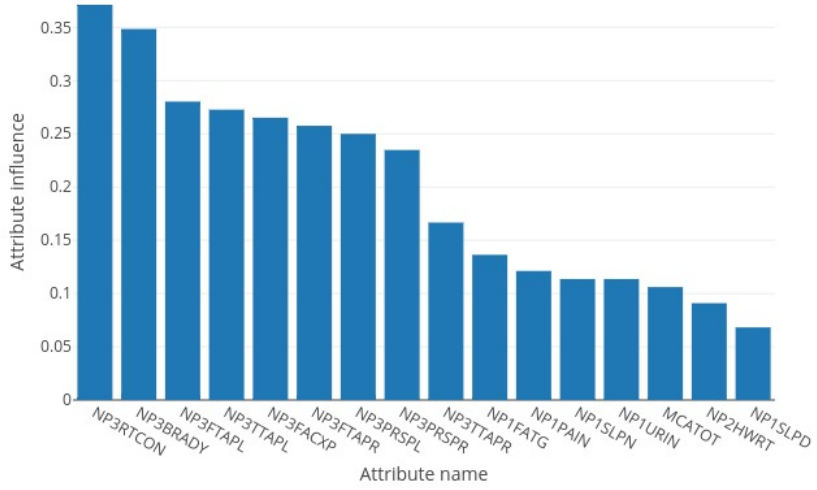


Fig. 3. Symptoms whose values improved most frequently when the overall status of patients improved. The acronyms are explained in Table 3.

that separate instances with different class values and punish the features that separate the instances with the same class value. If applied to our problem, these algorithms cannot take into account temporal progress of patients, i.e. they cannot track individual patients on their consecutive visits to the clinician. In effect, they show which attributes influence the initial assignment of patients into clusters, but reveal no information about attributes which are the most influential for changes of patients' overall status (i.e., for crossing of clusters). We note that the assignment to clusters was done based on the patients' overall status represented with sums of attributes values from the respective questionnaires presented in Section 3.1. Nevertheless, we evaluated the symptoms using the ReliefF algorithm [34, 35]. Out of the best 16 symptoms as evaluated by the ReliefF algorithm, 9 were selected into the top 16 most most influential symptoms (see Table 3) by Algorithm 1 (MCATOT, NP1SLPD, NP1URIN, NP3RTCON, NP1SLPN, NP1FATG, NP3PRSPL, NP3TTAPR, NP3TTAPL). Symptoms—such as bradykinesia—that are strong indicators of the disease progression were evaluated as insignificant by the ReliefF algorithm. For this reason, the results of ReliefF for symptom evaluation are not included.

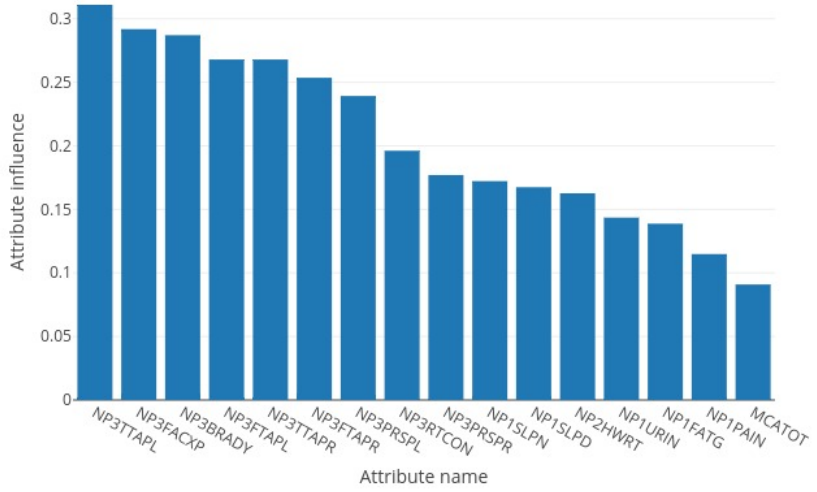


Fig. 4. Symptoms whose values worsen most frequently when the overall status of patients degraded. The acronyms are explained in Table 3.

5.1.1 Medical interpretation of the results

According to the consulting clinician, in general, the computed symptom importance is in accordance with the medical literature on Parkinson's disease [10, 1]. Below we present some further interesting findings.

Cognitive decline, as depicted by the MoCA total score, and bradykinesia are very important factors when considering changing patients' medications [10, 1]. Braykinesia is a score combining toe tapping [18, 14] (for lower limbs bradykinesia assessment), hand pronation/supination, and finger tapping (for upper limbs bradykinesia assessment) [21]. As confirmed by the expert, the constancy of rest tremor and pain are symptoms which are important for some patients who find these symptoms particularly bothersome and demand an intervention with medications. Dyskinesia and fluctuations are important symptoms not ranked at the top of the list according to our Algorithm 1. The reason is that the PPMI database includes many newly diagnosed and early-stage patients, for who these symptoms do not change values often.

The importance of handwriting is an interesting finding of the study and confirms recent studies [9] suggesting that handwriting could be a useful marker for disease diagnosis [36] and progression [27]. Our further analysis of patients with problematic handwriting revealed that these patients experience more prob-

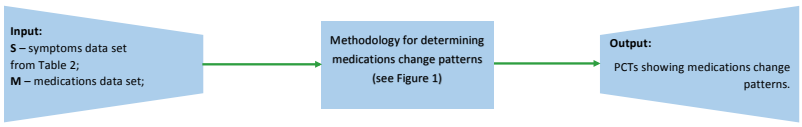


Fig. 5. A flowchart presenting the input, output, and method used for determining medications dosage change patterns detected from symptoms selected by clinicians. Details about the used symptoms data can be found in Table 2.

lems with their motor symptoms (reflected by the sum of symptoms from MDS-UPDRS Part III), and also suffer from bradykinesia, pain, and rigidity with higher severity than patients who do not have problems with handwriting. Results of our analysis also suggest that patient's handwriting sensitively reflects improvements and worsening of patients' motor symptoms.

5.2 Medications dosage change patterns detected from symptoms selected by clinicians

For this set of evaluations, we use the data set composed of symptoms selected by clinicians (see Section 3.3). A flowchart with the input, output, and method used in this experimental setting is presented in Figure 5. A pruned predictive clustering tree (PCT) model of medications changes based on the patient's status is shown in Figure 6. The PCT models the dosage changes of all three antiparkinson medication groups simultaneously, allowing for the detection of drug interactions based on the patient's status. Notice that in PCT construction, the user can decide how to prune the tree. In our experiments, we used the default pruning method, called C4.5 [39, 28].

The leaves of the predictive clustering tree hold information about the recorded therapy modifications. The components of the lists presented in each tree leaf predict dosage changes of levodopa, dopamine agonists, and MAO-B inhibitors, respectively. The list of numbers in the leaves represents the total number of instances that are described by the symptoms from the root to the leaf. The number of instances for which the proposed medications dosage change has actually happened are written in the square brackets. For example, the list [yes, yes, yes] presented in the first leaf on the left (Path 1), indicates that the dosages of levodopa, dopamine agonists, and MAO-B changed. The total number of covered instances is 20. Out of these 20 patients, for 17 the dosage of levodopa changed, for 12 the dosage of dopamine agonists changed and for 12 the dosage of MAO-B inhibitors changed.

The attributes of instances (patient-visit pairs) influencing this change are presented along the path from the tree root to the respective leaf. In this example, these are the patients who are younger than 65, have problems with ON/OFF fluctuations, and have problems with their impulsivity. The leaf [yes, no, no] on the right (Path 2) suggests that physicians only considered changes in levodopa. These dosage changes can be justified by the patients' symptoms,

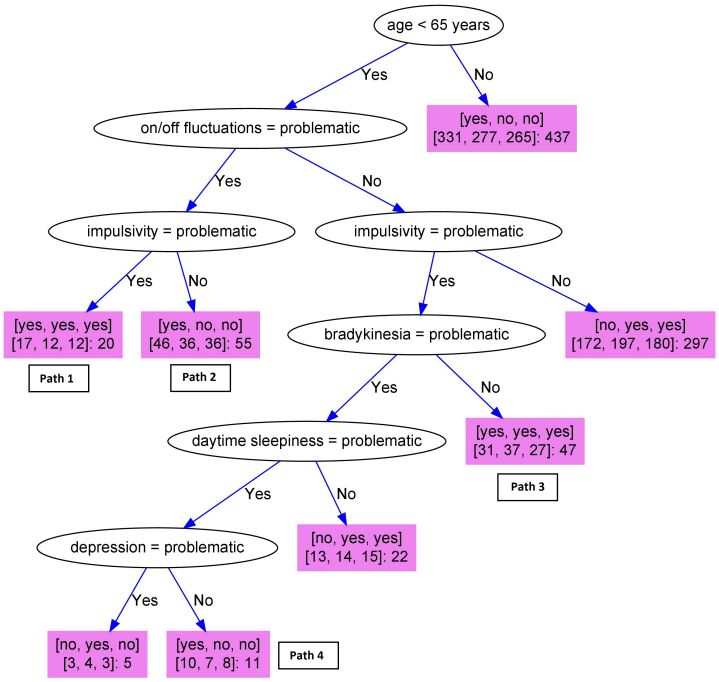


Fig. 6. Pruned predictive clustering tree modeling dosage changes for three groups of medications. Medication dosage changes are modeled by patients' symptoms.

i.e. patients have problems with ON/OFF fluctuations and no problems with their impulsivity. Moreover, in younger patients without ON/OFF fluctuation problems but with other problematic symptoms: impulsivity, bradykinesia and daytime sleepiness, the physicians also change only levodopa dosages (Path 4 in Figure 6). This might reflect the current clinical practice—many patients want treatment of their motor symptoms first, which usually improve based on increased levodopa dosages.

Path 4 in Figure 6 shows that for younger patients with problematic impulsivity and without problems with ON/OFF fluctuations and bradykinesia, physicians change the dosages of all three medication groups. These results are in accordance with the literature on Parkinson's disease [47] and were confirmed by the clinical expert.

The results reveal that if the patient experiences ON/OFF fluctuations problems (left subtree in Figure 6), physicians will react with the change of dosage of levodopa medications [12]. If the patients experience non-motor symptoms (e.g., impulsivity, depression), physicians will react by modifying the dosages of dopamine agonists [37]. This is in accordance with the literature on Parkinson's disease and was confirmed by the expert. Increased dosages of dopamine agonists can produce non-motor related side-effects. Physicians will react by lowering the

Table 4. Comparison of the classification accuracy obtained by the default model, the pruned multitask model, the pruned single-task models, and the random forest models.

Medications group	Default model	Multitask PCT model	Single-task PCT model	Random forest ensembles
Levodopa	0.637	0.685	0.686	0.695
Dopamine agonists	0.501	0.642	0.642	0.622
MAO-B	0.518	0.602	0.586	0.601

dosage of dopamine agonists (consequently increasing the dosage of levodopa). This was revealed in our post analysis, where we followed the actual changes of levodopa and dopamine agonists. In this post analysis the target variables (levodopa and dopamine agonists) had three values: *increase*, *decrease*, and *unchange*. The PCT model was built on the symptoms data presented in Section 3.3 in combination with the newly generated target features.

While prediction is not the ultimate goal of the developed methodology, reasonably high classification accuracy on a separate data set can increase clinicians' trust in using the model. Table 4 presents the classification accuracy of four models: i) the default model (predicting the most probable value for each target), ii) the multitask PCT classification model, iii) the single-task decision tree models constructed separately for each medications group, and iv) the multitask random forest model [5, 20]. The results are obtained using 10-fold cross-validation. The results show that random forests generate models that have slightly better classification accuracy for levodopa. However, the PCT models yield better classification accuracy for dopamine agonists. Multitask PCT model and random forests return comparable classification accuracy for MAO-B. The advantage of using the multitask tree approach is the ability to observe the interactions between the targets.

We employed the Wilcoxon [49] paired test to examine whether there are statistical differences between the performance of the multitask approach and the single task approach, and the multitask approach and random forest. Results showed that there are no statistical differences at the level of significance α=0.05 for any pair of the above pairs. We used the same folds across all approaches.

5.2.1 Impact of symptoms history

We analyzed the temporal aspect of the proposed approach. Our data set offers certain time-related information (1-4 observations are available for each patient in the data set, one for each visit). So far we only analyzed the changes in symptoms and dosage between two consecutive visits. According to our consulted clinicians, the current state of the patient is all that matters to the clinician when considering their therapy, so this makes sense. However, the question remains if taking into account more than one historical event can improve models.

To adequately answer this comment, we conducted a separate set of experiments. We looked further back into patients' history to see how their medications

have changed, based on symptoms from more distant visits. Comparison of 10-fold cross validation classification accuracy on models with different spans of look-back showed that the classification accuracy of models decreases as we include references to more distant visits. The highest accuracy is achieved when we consider only the actual state of the patients (these results are presented in this section).

5.3 Medications Dosage Change Patterns Detected from Extended Symptoms Data

The PCT model from Section 5.2 was generated on a set of symptoms selected based on the expert's choice. In this Section, we explore the model for dosage change of antiparkinson medications if in addition to the symptoms that are pre-selected by the clinicians we include also the most influential attributes from Table 3. We present the description of the newly introduced symptoms, the predictive clustering tree model generated on this extended data set, short interpretation of the tree, the classification accuracy of the models, and a short discussion on the differences between the original model (Section 5.2) and the revised model (Section 5.3.2).

5.3.1 Extended data set

As already mentioned, when monitoring the patient's status and deciding about the modification of their medications therapy, clinicians think in terms whether the symptom's severity is *normal* for a Parkinson's disease patient or it is *problematic* and a change of dosage of antiparkinson medications is needed. Table 5 presents the most influential *motor* and *non-motor* symptoms according to Algorithm 1 and the intervals for their quantization into *normal* and *problematic* symptoms values. Six of the symptoms from Table 3 were merged into three new (revised) symptoms. These six symptoms were pairs of three underlying symptoms, each concerning a different side of the body (left or right), and were therefore paired into three new symptoms. The three new symptoms are: toe tapping, finger tapping, and hand pronation/supination. The values of the newly constructed symptoms are obtained as the maximum of the two basic symptoms values (left and right).

The extended symptoms data set used in the experiments below is presented in Table 6. These symptoms are *motor*, *non-motor*, and *epidemilogical*, consisting of the symptoms that were pre-selected by our consulting expert (see Section 3.3) and the most influential symptoms returned by Algorithm 1 (Table 5). We decided to omit the *cognitive disorder* attribute due to the fact that its only values present in the database were *normal* and missing. The reason for this is that in this analysis we only consider patients with included medications data.

Note that out of the 16 symptoms that were top-ranked by Algorithm 1, our consulting clinician reported 3 as the symptoms they consider when deciding about the change of Parkinson's disease patient's therapy. These symptoms are *cognition*, *daytime sleepiness*, and *bradykinesia* (marked in bold in Table 5).

Table 5. Description of the *motor* (upper part) and *non-motor* (lower part) symptoms which are reported as the most influential by Algorithm 1. Toe tapping, finger tapping, and hand pronation/supination were generated as the maximum value of the basic symptom on the patient's left and right side. The values intervals (*normal* and *problematic*) were defined by the clinician. The three symptoms marked with bold typeface were independently selected by clinicians as the most important.

Symptom	Data set	Question number	Normal values interval	Problematic values interval
bradykinesia	MDS-UPDRS Part III	3.14	0–1	2–4
toe tapping	MDS-UPDRS Part III	max(3.7a, 3.7b)	0–1	2–4
constancy of rest	MDS-UPDRS Part III	3.18	0–1	2–4
facial expression	MDS-UPDRS Part III	3.2	0–1	2–4
finger tapping	MDS-UPDRS Part III	max(3.4a, 3.4b)	0–1	2–4
hand pronation/supination	MDS-UPDRS Part III	max(3.6a, 3.6b)	0–1	2–4
sleep problems	MDS-UPDRS Part Ip	1.7	0–1	2–4
daytime sleepiness	MDS-UPDRS Part I	1.8	0–1	2–4
handwriting	MDS-UPDRS Part II	2.7	0–1	2–4
fatigue	MDS-UPDRS Part Ip	1.13	0–1	2–4
urinary problems	MDS-UPDRS Part Ip	1.10	0–1	2–4
pain and other sensations	MDS-UPDRS Part Ip	1.9	0–1	2–4
cognitive disorder	MoCA	SUM	26–30	<26

Table 6. Extended symptoms data set consisting of symptoms handpicked by the expert and the most influential symptoms ranked by Algorithm 1. Details about the symptoms can be found in Tables 2 and 5.

Motor symptoms	Non-motor symptoms	Epidemilogical symptoms
bradykinesia	*daytime sleepiness*	*age*
tremor	*impulsivity*	*disease duration*
gait	*depression*	
dyskinesia	*hallucinations*	
ON/OFF fluctuations	*sleep problems*	
toe tapping	*handwriting*	
constancy of rest	*fatigue*	
facial expression	*urinary problems*	
finger tapping	*pain and other sensations*	
hand pronation/supination		

5.3.2 Revised results and discussion

The extended symptoms data set was used as an input to our methodology for determining medications dosage change patterns in PPMI data using predictive clustering trees (presented in Section 4.2). A flowchart outlining the input, output, and method used in this experimental setting is presented in Figure 7. The obtained model for symptoms scenarios that caused clinicians reaction with medications dosage change is presented in Figure 8.

The revised model for dosage changes is slightly different from the original model presented in Figure 6. The roots of the trees are the same, i.e. the clinician's decision about modifying the patient's medications treatment is mostly influenced by the age of the patient. For younger patients, the decision is influ-

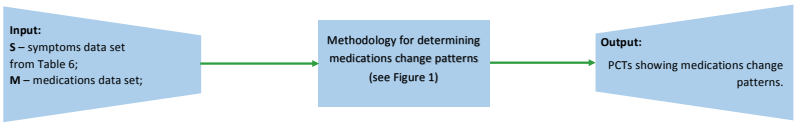


Fig. 7. A flowchart presenting the input, output, and method used for determining patterns of medications dosage change from the extended symptoms data. The extended symptoms data set is presented in Table 6.

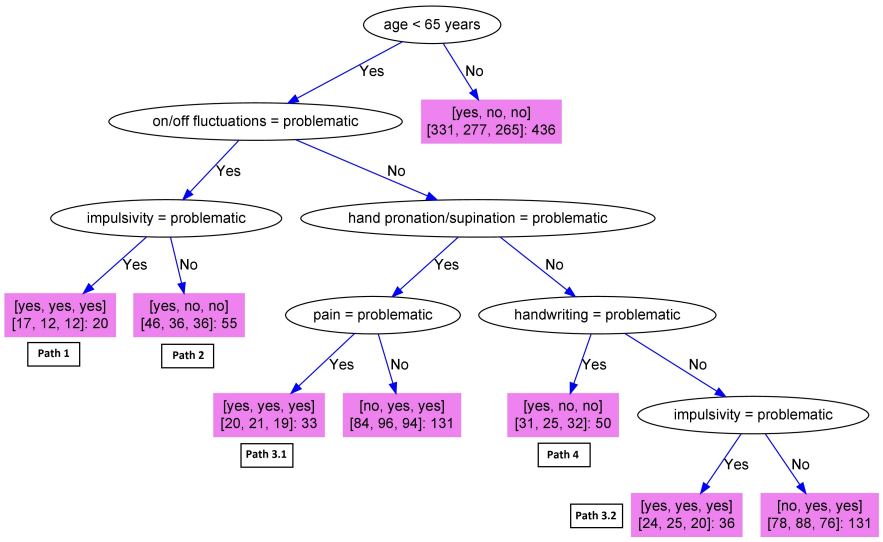


Fig. 8. Pruned predictive clustering tree modeling the dosage changes for three groups of medications. The model is generated on the extended set of Parkinson's disease patients symptoms. For improved model readability, the minimal number of covered instances is set to 20.

enced also by their on/off fluctuations and impulsivity. The right-hand side of the subtree concerning younger patients is different. In this subtree, the symptoms that influence the dosage change of antiparkinson medications are the newly introduced symptoms: pain and other sensations, hand pronation/supination, and handwriting. Path 1 and Path 2 are the same in both models. Path 4 in both models reveals medications change pattern [yes, no, no], indicating that based on the symptoms patterns (the paths from the root to the leaf of the tree) the clinicians consider changing the dosage of levodopa, and leave the dosages of dopamine agonists and MAO-B inhibitors unchanged. Paths 3.1 and 3.2 suggest that the clinician should consider updating the dosages of all antiparkinson medications which is similar to Path 3 of Figure 6.

For each of the medications groups, Table 7 presents the classification accuracy of the default model, revised multitask PCT model, revised single-task PCT

Table 7. Comparison of the classification accuracy obtained by the default model, the pruned multitask PCT model, the pruned single-task PCT models, and the random forest ensemble model on the extended symptoms data set.

Medications group	Default model	Multitask PCT model	Single-task PCT model	Random forest ensemble
Levodopa	0.637	0.657	0.671	0.683
Dopamine agonists	0.501	0.631	0.630	0.642
MAO-B	0.518	0.583	0.572	0.615

models, and random forest multitask model. Results are obtained using 10-fold cross-validation. As it is the case with the model from Figure 6, the accuracy values obtained by the multitask PCT model are comparable to those obtained by the single-task PCT model and are better when compared to the default model. The multitask random forest ensemble returned the best classification accuracy for all targets. This model also has an improved classification accuracy for dopamine agonists and MAO-B compared to models from Section 5.2 (see Figure 6 and Table 4). This improvement can be explained with the additional information available in the extended set of attributes and non-trivial interactions between different targets, which can be captured by ensembles. The main disadvantage of the ensemble multitask models is their lack of model interpretability. The Wilcoxon [49] paired test revealed that for the target variable levodopa the random forest ensemble performs significantly better than our multitask approach (α=0.05, p-value=0.012). For the target variable MAO-B, the multitask approach performs significantly better than the single-task approach (α=0.05, p-value=0.018). Other differences were not significant. We used the same folds across approaches.

The classification accuracy of both the revised multitask PCT model and the revised single-task PCT models are lower than the accuracies of the models generated on the original symptoms data set (Table 2). A reason for this difference may be the fact that our models are trained on and reflect the history of clinicians' decisions, and do not necessarily reflect the actual symptoms clinicians should react to.

5.3.3 Medical evaluation of the results

For patients covered by rules from Path 1 and Path 2 it is reasonable to introduce levodopa and try to provide the optimal dosage even in younger patients (average age of 53 years) when they have on/off fluctuations (i.e. disease is rapidly progressing). The presence of impulsivity dictates the medications dosage changes the clinician should make. Path 3.1 covers younger patients (average age of 52.42 years) who suffer from severe bradykinesia (NP3BRADY $= 1.94 \pm 0.84$[6]). Their overall motor symptoms are severe, i.e. the sum of MDS-UPDRS Part III

[6] In further analysis of the tree leaves in the model from Figure 8 we calculated average symptom values of covered patients.

(NP3SUM) is 34.03 ± 11.99. Patients' quality of daily living is affected, i.e. the sum of MDS-UPDRS Part II (NP2SUM) is 13.85 ± 6.36. Along with the presence of pain, many changes in medications dosages are done in an effort to better manage the advanced disease severity. Patients who do not have problems with pain and are treated with [no,yes,yes] medications dosage change pattern are patients who also have severe motor symptoms (NP3SUM = 31.06 ± 0.84 and disturbing bradykinesia (NP3BRADY = 1.71 ± 0.84)). However, their overall status is slightly better and the mild problems with pain lead to more dosage changes of dopamine agonists and MAO-B inhibitors, and a stable treatment with levodopa.

Patients covered by Path 4 are overall in a better condition that patients mentioned in previous paths. Their motor symptoms are less severe (NP3SUM = 20.39 ± 9.15), they do not have problems with on/off fluctuations, they have mild bradykinesia, and have no cognitive problems. Their handwriting seems to be a useful marker of disease progression which leads to dosage changes in levodopa. Changes of medication dosages for patients covered by Path 3.2 are imposed by the problematic impulsivity. Dosages of dopamine agonists are lowered to stabilize impulsivity, while levodopa is increased in order to control the motor symptoms. Younger patients who do not have problems with impulsivity (nor problems with on/off fluctuations, hand pronation/supination, handwriting) and are treated with [no,yes,yes] medications dosage change pattern are patients who are in better condition than all the other patients included in the predictive clustering tree from Figure 8. Reasonably, only dopamine agonists and MAO-B inhibitors are modified in an effort for better management of the disease. Levodopa is either not prescribed or only low dosages are prescribed. Older patients (average age of 68.45 ± 4.87 years) have problems with many symptoms. The disease is managed with levodopa, and an optimal regime is sought through changes.

6 Conclusions

We present the methodology to detect trigger symptoms for change of medications therapy of Parkinson's disease patients. We consider trigger symptoms to be the ones which press the physicians to make modifications of the treatment for their patients. We test the developed methodology on a chosen subset of time-stamped PPMI data. The data set offers an insight into the patients' symptoms progression through time, as well as the response of physicians following problematic states of either motor or non-motor symptoms. We identify clinically confirmed patients' symptoms indicating the need for medication changes.

The proposed approach allows identifying patient subgroups for which certain medications modifications have either a positive or a negative effect. By post analysis of the patients who respond well to the medications modification and those who do not, and the underlying characteristics of each group, we may be able to assist the physicians with the therapy modifications for a given patient by narrowing the number of possible medication prescriptions scenarios.

We also present an algorithm for determining the symptoms which have the largest influence on the change of the Parkinson's disease patients' overall status. These are the symptoms that change most frequently as the status of the patient improves/declines. We relate this work with our previous work, where we developed a methodology for determining groups of patients with similar severity of symptoms and establishing how the disease progresses in terms of the severity of several groups of symptoms.

Our results show that some of the most impactful symptoms for changes in the patients overall status detected by Algorithm 1, are currently not considered by the clinicians when deciding about the change of antiparkinson medication dosages. This requires further study and offers an opportunity for improved disease management in the future.

In future work, we plan to apply model explanation approaches to describe relevant subgroups of patients, therapy change patterns with a positive influence on the control of symptoms, and therapy patterns which are more likely to lead to side effects. There are some open opportunities in the analysis of more than one previous time point. Taking longer history into account we might be able to detect groups of patients which do not react well to the changes in antiparkinson medications.

Acknowledgements. This work was supported by the PD_manager project, funded within the EU Framework Programme for Research and Innovation Horizon 2020 grant 643706. We acknowledge the financial support from the Slovenian Research Agency (research core fundings No. P2-0209 and P2-0103). This research has received funding also from the European Unions Horizon 2020 research and innovation programme under grant agreement No. 720270 (HBP SGA1).

Data used in the preparation of this article were obtained from the Parkinsons Progression Markers Initiative (PPMI) (`www.ppmi-info.org/data`). For up-to-date information on the study, visit `www.ppmi-info.org`. PPMI—a public-private partnership—funded by the Michael J. Fox Foundation for Parkinson's Research and funding partners. Corporate Funding Partners are: AbbVie, Avid Radiopharmaceuticals, Biogen, BioLegend, Bristol-Myers Squibb, GE Healthcare, GLAXOSMITHKLINE (GSK), Eli Lilly and Company, Lundbeck, Merck, Meso Scale Discovery (MSD), Pfizer Inc, Piramal Imaging, Roche, Sanofi Genzyme, Servier, Takeda, Teva, UCB. Philanthropic Funding Partners: Golub Capital. List of funding partners is found at `www.ppmi-info.org/fundingpartners`.

References

[1] Evidence Based Medicine Publications for Treatment of Motor and Non-motor symptoms of Parkinson's disease. `http://www.movementdisorders.org/MDS/Resources/Publications-Reviews/EBM-Reviews.htm`. Accessed: 2017/10/20.

[2] A. Arauzo-Azofra, J. L. Aznarte, and J. M. Bentez. Empirical study of feature selection methods based on individual feature evaluation for classification problems. *Expert Systems with Applications*, 38(7):8170 – 8177, 2011.

[3] H. Blockeel and L. De Raedt. Top-down induction of first-order logical decision trees. *Artificial Intelligence*, 101(1-2):285–297, 1998.
[4] 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)*, pages 55–63, 1998.
[5] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
[6] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.
[7] R. Caruana, S. Baluja, and T. Mitchell. Using the future to "sort out" the present: Rankprop and multitask learning for medical risk evaluation. In *Advances in Neural Information Processing Systems*, pages 959–965, 1996.
[8] J. Dalrymple-Alford, M. MacAskill, C. Nakas, L. Livingston, C. Graham, G. Crucian, T. Melzer, J. Kirwan, R. Keenan, S. Wells, et al. The MoCA: well-suited screen for cognitive impairment in Parkinson disease. *Neurology*, 75(19):1717–1725, 2010.
[9] P. Drotár, J. Mekyska, I. Rektorová, L. Masarová, Z. Smékal, and M. Faundez-Zanuy. Analysis of in-air movement in handwriting: A novel marker for Parkinson's disease. *Computer Methods and Programs in Biomedicine*, 117(3):405–411, 2014.
[10] J. Ferreira, R. Katzenschlager, B. Bloem, U. Bonuccelli, D. Burn, G. Deuschl, E. Dietrichs, G. Fabbrini, A. Friedman, P. Kanovsky, et al. Summary of the recommendations of the EFNS/MDS-ES review on therapeutic management of Parkinson's disease. *European Journal of Neurology*, 20(1):5–15, 2013.
[11] T. Foltynie, C. Brayne, and R. A. Barker. The heterogeneity of idiopathic Parkinson's disease. *Journal of Neurology*, 249(2):138–145, 2002.
[12] S. H. Fox, R. Katzenschlager, S.-Y. Lim, B. Ravina, K. Seppi, M. Coelho, W. Poewe, O. Rascol, C. G. Goetz, and C. Sampaio. The movement disorder society evidence-based medicine review update: Treatments for the motor symptoms of Parkinson's disease. *Movement Disorders*, 26(S3):S2–S41, 2011.
[13] D. Gil and M. Johnson. Diagnosing Parkinson by using artificial neural networks and support vector machines. *Global Journal of Computer Science and Technology*, 9(4):63–71, 2009.
[14] C. G. Goetz, B. C. Tilley, S. R. Shaftman, G. T. Stebbins, S. Fahn, P. Martinez-Martin, W. Poewe, C. Sampaio, M. B. Stern, R. Dodel, et al. Movement Disorder Society-sponsored revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS): Scale presentation and clinimetric testing results. *Movement Disorders*, 23(15):2129–2170, 2008.
[15] M. M. Hoehn and M. D. Yahr. Parkinsonism onset, progression, and mortality. *Neurology*, 17(5):427–427, 1967.
[16] A. Holzinger. Trends in interactive knowledge discovery for personalized medicine: Cognitive science meets machine learning. *IEEE Intelligent Informatics Bulletin*, 15(1):6–14, 2014.
[17] A. J. Hughes, S. E. Daniel, L. Kilford, and A. J. Lees. Accuracy of clinical diagnosis of idiopathic Parkinson's disease: A clinico-pathological study of 100 cases. *Journal of Neurology, Neurosurgery & Psychiatry*, 55(3):181–184, 1992.
[18] J.-W. Kim, Y. Kwon, Y.-M. Kim, H.-Y. Chung, G.-M. Eom, J.-H. Jun, J.-W. Lee, S.-B. Koh, B. K. Park, and D.-K. Kwon. Analysis of lower limb bradykinesia in Parkinson's disease patients. *Geriatrics & Gerontology International*, 12(2):257–264, 2012.
[19] K. Kira and L. A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, volume 2, pages 129–134, 1992.

[20] D. Kocev, C. Vens, J. Struyf, and S. Džeroski. Ensembles of multi-objective decision trees. In *European Conference on Machine Learning*, pages 624–631. Springer, 2007.

[21] C. Lainscsek, P. Rowat, L. Schettino, D. Lee, D. Song, C. Letellier, and H. Poizner. Finger tapping movements of Parkinson's disease patients automatically rated using nonlinear delay differential equations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(1):013119, 2012.

[22] S. Lewis, T. Foltynie, A. Blackwell, T. Robbins, A. Owen, and R. Barker. Heterogeneity of Parkinson's disease in the early clinical stages using a data driven approach. *Journal of Neurology, Neurosurgery & Psychiatry*, 76(3):343–348, 2005.

[23] L.-Y. Ma, P. Chan, Z.-Q. Gu, F.-F. Li, and T. Feng. Heterogeneity among patients with Parkinson's disease: Cluster analysis and genetic association. *Journal of the Neurological Sciences*, 351(1):41–45, 2015.

[24] K. Marek, D. Jennings, S. Lasch, A. Siderowf, C. Tanner, T. Simuni, C. Coffey, K. Kieburtz, E. Flagg, S. Chowdhury, et al. The Parkinson's Progression Markers Initiative (PPMI). *Progress in Neurobiology*, 95(4):629–635, 2011.

[25] National Collaborating Centre for Chronic Conditions. *Parkinson's disease: National clinical guideline for diagnosis and management in primary and secondary care*. London: Royal College of Physicians, 2006.

[26] S. Patel, K. Lorincz, R. Hughes, N. Huggins, J. Growdon, D. Standaert, M. Akay, J. Dy, M. Welsh, and P. Bonato. Monitoring motor fluctuations in patients with Parkinson's disease using wearable sensors. *IEEE Transactions on Information Technology in Biomedicine*, 13(6):864–873, 2009.

[27] S. Pinto and J.-L. Velay. Handwriting as a marker for PD progression: a shift in paradigm. *Neurodegenerative Disease Management*, 5(5):367–369, 2015.

[28] J. R. Quinlan. *C4. 5: Programs for machine learning*. Elsevier, 2014.

[29] R. G. Ramani and G. Sivagami. Parkinson disease classification using data mining algorithms. *International Journal of Computer Applications*, 32(9):17–22, 2011.

[30] B. Rana, A. Juneja, M. Saxena, S. Gudwani, S. S. Kumaran, M. Behari, and R. Agrawal. Graph-theory-based spectral feature selection for computer aided diagnosis of Parkinson's disease using T1-weighted MRI. *International Journal of Imaging Systems and Technology*, 25(3):245–255, 2015.

[31] J. Reijnders, U. Ehrt, R. Lousberg, D. Aarsland, and A. Leentjens. The association between motor subtypes and psychopathology in Parkinson's disease. *Parkinsonism & Related Disorders*, 15(5):379–382, 2009.

[32] O. Reyes, C. Morell, and S. Ventura. Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context. *Neurocomputing*, 161:168–182, 2015.

[33] C. N. Riviere, S. G. Reich, and N. V. Thakor. Adaptive Fourier modeling for quantification of tremor. *Journal of Neuroscience Methods*, 74(1):77–87, 1997.

[34] M. Robnik-Šikonja and I. Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine learning*, 53(1-2):23–69, 2003.

[35] M. Robnik-Šikonja and P. Savicky. Corelearn: Classification, regression and feature evaluation, r package version 0.9. 45 (2015). *URL http://CRAN. R-project. org/package= CORElearn*.

[36] S. Rosenblum, M. Samuel, S. Zlotnik, I. Erikh, and I. Schlesinger. Handwriting as an objective tool for Parkinson's disease diagnosis. *Journal of Neurology*, 260(9):2357–2361, 2013.

[37] K. Seppi, D. Weintraub, M. Coelho, S. Perez-Lloret, S. H. Fox, R. Katzenschlager, E.-M. Hametner, W. Poewe, O. Rascol, C. G. Goetz, et al. The movement disorder

society evidence-based medicine review update: Treatments for the non-motor symptoms of Parkinson's disease. *Movement Disorders*, 26(S3), 2011.

[38] R. Shang, W. Wang, R. Stolkin, and L. Jiao. Subspace learning-based graph regularized feature selection. *Knowledge-Based Systems*, 112:152–165, 2016.

[39] J. Struyf, B. Zenko, H. Blockeel, C. Vens, and S. Dzeroski. Clus: User's manual, 2010.

[40] X. Sun, Y. Liu, J. Li, J. Zhu, X. Liu, and H. Chen. Using cooperative game theory to optimize the feature selection problem. *Neurocomputing*, 97:86–93, 2012.

[41] J. Timmer, C. Gantert, G. Deuschl, and J. Honerkamp. Characteristics of hand tremor time series. *Biological Cybernetics*, 70(1):75–80, 1993.

[42] A. Valmarska, D. Miljkovic, S. Konitsiotis, D. Gatsios, N. Lavrač, and M. Robnik-Šikonja. Combining multitask learning and short time series analysis in Parkinson's disease patients stratification. In *Proceedings of the Conference on Artificial Intelligence in Medicine in Europe*, pages 116–125. Springer, 2017.

[43] A. Valmarska, D. Miljkovic, N. Lavrač, and M. Robnik-Šikonja. Analysis of medications change in Parkinson's disease progression data. *Journal of Intelligent Information Systems*, 2018.

[44] A. Valmarska, D. Miljkovic, M. Robnik-Šikonja, and N. Lavrač. Multi-view approach to Parkinson's disease quality of life data analysis. In *Proceedings of the International Workshop on New Frontiers in Mining Complex Patterns*, pages 163–178. Springer, 2016.

[45] M. Visser, J. Marinus, A. M. Stiggelbout, and J. J. Van Hilten. Assessment of autonomic dysfunction in Parkinson's disease: The SCOPA-AUT. *Movement Disorders*, 19(11):1306–1312, 2004.

[46] R. A. Washburn, K. W. Smith, A. M. Jette, and C. A. Janney. The physical activity scale for the elderly (PASE): development and evaluation. *Journal of Clinical Epidemiology*, 46(2):153–162, 1993.

[47] D. Weintraub, J. Koester, M. N. Potenza, A. D. Siderowf, M. Stacy, V. Voon, J. Whetteckey, G. R. Wunderlich, and A. E. Lang. Impulse control disorders in Parkinson disease: a cross-sectional study of 3090 patients. *Archives of Neurology*, 67(5):589–595, 2010.

[48] D. Weintraub, E. Mamikonyan, K. Papay, J. A. Shea, S. X. Xie, and A. Siderowf. Questionnaire for impulsive-compulsive disorders in Parkinson's disease–rating scale. *Movement Disorders*, 27(2):242–247, 2012.

[49] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[50] D. Zhang, D. Shen, and ADNI. Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in Alzheimer's disease. *NeuroImage*, 59(2):895–907, 2012.

[51] J. Zhou, J. Liu, V. A. Narayan, J. Ye, and ADNI. Modeling disease progression via multi-task learning. *NeuroImage*, 78:233–248, 2013.

Chapter 7

Conclusions

Our work provides novel approaches to descriptive data mining of Parkinson's disease data. We address Parkinson's disease progression and therapy modification. In this chapter, we provide a summary of the hypotheses from Chapter 1 and the scientific contributions covered in Chapters 4, 5 and 6. We conclude our work with ideas for future work.

7.1 Summary of Contributions

The thesis presents improvements in three research problems related to Parkinson's disease progression management.

1. *Development of improved algorithms for classification rule learning and subgroup discovery.* We examined the advantage of separating the refinement and selection phase of the rule learning process. For that purpose, in Chapter 4 we introduced two new beam search algorithms for rule learning, one for subgroup discovery (DoubleBeam-SD) and one for classification rule learning (DoubleBeam-RL). Both algorithms employ a two-step phase separation: i) they use separate beams for the refinement and selection phase of rule learning and ii) they offer the possibility to use separate heuristics for rule refinement and rule selection. The deployment of separate heuristics exploiting specifics of the refinement and selection phase and the utilization of separate beams widens the search space and can construct rules which might otherwise not have been found by the current state-of-the-art algorithms.

   We compared the performance of the DoubleBeam-SD algorithm and the DoubleBeam-RL to their state-of-the-art counterparts. Experiments on 20 UCI data sets showed that the performance of each of the considered algorithms depends on their parameters. In the classification rule learning, the experimental results confirm previously shown benefits of using two separate heuristics for rule refinement and rule selection—comparable or increased predictive power and rules with better descriptive power. In subgroup discovery, DoubleBeam-SD algorithm variants outperform several state-of-the-art related algorithms.

2. *Development of a methodology for Parkinson's disease progression.* We developed a methodology for the analysis of short time series Parkinson's disease data. The methodology is based on clustering of temporal symptoms data (Chapter 4). We show that patients are divided into groups which can be characterized with similar symptoms, i.e. their overall status. The results show that these groups of patients can be ordered (either totally or partially) in accordance with the severity of the symptoms indicating the patients' overall status. Since the status of patients is

bound to change through time due to both the natural progression of the disease as well as the antiparkinson therapies, the patients change cluster membership between time points, thus indicating the improvement or degradation of their status.

As a starting point for determining frequent patterns of disease progression, we use patient's assignment to clusters. Since the number of available time points for each patient is limited, we adapted the skip-grams approach to obtain more robust and more reliable patterns of disease progression. Post analysis showed that patients that share similar patterns of disease progression also share etymological, motor, and non-motor symptoms.

3. *Development of a methodology for detection of medications changes.* We presented a methodology for detecting patterns of medications dosage changes based on the patients' status (Chapter 6). We introduced a predictive clustering trees (PCT)-based methodology for detection of medications changes. The generated models serve three purposes: i) determine which symptoms influence the clinicians to modify the patients' therapies, ii) determine patterns of antiparkinson medications dosage changes in response to the patients' symptoms, and iii) identify groups of patients that are similar in regards to both the symptoms they are experiencing as well as the prescribed therapy modifications.

   The experiments show that the multitask models exhibit comparable or better predictive performance in comparison to the respective single task models, with the added advantage of simultaneously assessing changes in patients' therapies. The identification of patterns in therapy modifications and groups of similar patients is a step closer to a more personalized treatment of Parkinson's disease patients. The methodology can be adapted to other areas where the status conditioned on actual changes need to be examined. The methodology can be used as a foundation for building a decision support system, where the expert knowledge and the knowledge obtained from our methodology are merged. We also present the algorithm which detects the symptoms with the strongest impact on significant changes in the patients' overall status.

7.2 Summary of Hypotheses Confirmations

In Section 1.3 we hypothesized that the separation of the two phases of the rule learning process will lead towards the classification rule learning algorithms and the subgroup discovery algorithms generating rules with the improved quality compared to the rules produced by their state-of-the-art counterparts. We also hypothesized that the introduction of two beams and separate heuristics for each of the phases in the rule learning process will widen the search space and enable the algorithms to encounter rules which would otherwise be missed by standard algorithms. We address these issues in the publication included in Section 4.2. We present two new algorithms for subgroup discovery and rule learning, and we empirically show that the two-fold separation of the rule learning phases can widen the search place and thus enable the new algorithms to generate rules which would have otherwise been missed. The hypothesis was partly confirmed since the rules generated by our algorithms are comparable and in some cases better than the rules generated by the state-of-the-art algorithms used in the comparison.

Our second hypothesis was that Parkinson's disease patients can be divided into groups of patients with similar symptoms. These groups of patients can be partially ordered according to the severity of the symptoms describing each of the clusters. We address these issues in the publication included in Section 5.2. The hypothesis was confirmed. We

empirically show that the patients' status will change through time and will be reflected through patients' membership in different groups at separate time points. We show that the clusters are at least partially ordered, and the transition between two clusters can indicate a transition in the overall status of the patient, thus signaling an improvement or a deterioration of the patients' quality of life. Using skip-grams we were able to discover robust patterns of disease progression based on the patients' change of cluster assignments.

Our third hypothesis was that by using multitask approach we will be able to determine symptoms that trigger changes in medication dosages. The hypothesis was confirmed. In the published publication from Section 6.2 we empirically show that therapy dosage modifications are the result of the patients' status reflected by the clinical evaluation of their symptoms. We also include the consulting clinicians' interpretation of modifications of patients' therapies as well as references to the medical literature for treatment of Parkinson's disease patients.

7.3 Strenghts and Limitations of the Developed Approaches

In this section, we analyze the strengths of the algorithms and methodologies presented in the thesis, and critically evaluate their limitations.

7.3.1 Strenghts

The developed algorithms and methodologies are directed towards improved methods for the analysis of Parkinson's disease progression data, emphasizing the importance of using descriptive data mining methods for medical data analysis for a more personalized care of patients.

- The DoubleBeam-SD and DoubleBeam-RL algorithms exploit the use of separate heuristics for the refinement and selection phase in the rule learning process.
- The introduction of two beams and separate heuristics for each phase widens the search space thus allowing for the discovery of good rules which might have otherwise been overlooked.
- Side effects of the presented inverted heuristics are longer rules which may offer higher descriptive power to the built models.
- DoubleBeam-RL and DoubleBeam-SD algorithms are publicly available on GitHub and in the ClowdFlows platform which allows for the repeatability of experiments.
- The proposed methodologies address an important issue of determining patterns of disease progression and medication therapy modifications in order to keep the patients' status stable.
- The methodology for determining disease progression patterns using skip-grams preserves the temporal nature of patients' data which is not the case when using other methods for pattern detection such as itemsets from association rule learning.
- The methodologies for the analysis of Parkinson's disease progression and medications dosage changes preserve the individuality of each patient. Patterns are determined based on the changes in the overall status or in therapy modifications of each patient separately.

7.3.2 Limitations

Users of the algorithms and methodologies presented in the thesis should be aware of the following issues:

- The DoubleBeam-SD and DoubleBeam-RL algorithms do not employ pruning techniques. The increased search space can lead to overfitting. Users can use a separate validation data set to select the optimal size of the rules.
- The DoubleBeam-SD and DoubleBeam-RL algorithms are unable to handle big data.
- The performance of the DoubleBeam-RL algorithm and the DoubleBeam-SD algorithm depends on their parameters. As a start, the users can use the default parameters from Chapter 4. We suggest using cross-validation to determine optimal parameter values.
- Patterns of disease progression are based on clustering computed on the aggregate data presenting the overall status of Parkinson's disease patients. Even though Parkinson's disease clinicians often use this approach as a validator of the overall status of their patients (Goetz et al., 2008), the approach does not reveal the actual symptoms affecting the change of patients' status over time. A possible solution are better multi-view clustering approaches.
- Discovered patterns of medications change are overly general. We obtain them with a *yes/no* model indicating whether a change in the LEDD value of a particular antiparkinson medication has occurred or not. Further investigation of the models and instances covered by each pattern can reveal what are the actual medications dosage changes that have occurred. This is a time-consuming process which is not yet automated.
- The methodology is able to handle only preprocessed PPMI data and cannot handle data from wearable sensors of Parkinson's disease patients. The performance of our methodologies depends on the quality of the used data, which can be subjectively influenced by both the clinicians and the patients when evaluating the symptoms. Even though the clinicians follow the guidelines for the evaluation of Parkinson's disease patients, the final mark is subject to their own personal decision.
- The results are evaluated by consulting clinicians and either confirmed or refused with references from the Parkinson's disease medical literature. The disadvantage of this approach is that the clinicians may confirm or deny findings based on their experience and knowledge, and not based on objective criteria.
- The methods developed in Chapters 5 and 6 are not made publicly available due to their close dependence and integration with the data sets used. The interested reader is welcome to contact the author.

7.4 Further Work

The three main research contributions described in Chapters 4, 5, and 6 show promising results and deserve further investigation.

The newly developed algorithms for rule learning and subgroup discovery (DoubleBeam-RL and DoubleBeam-SD) demand further research in terms of stopping criteria and rule pruning heuristics, e.g., a post-processing rule pruning similar to the one available in Ripper. In the subgroup discovery setting, experimental results showed the advantage of using

WRACC over the traditional rule learning heuristics in obtaining interesting subgroups. Given the interest of clinicians in detecting interesting subgroups of patients, the development of new heuristics specialized for the detection of interesting subgroups is a promising research path which should be explored in the future. An interesting approach is to evaluate the performance of *lift*, the heuristic used in the Hedwig system (Vavpetič et al., 2013) for semantic subgroup discovery.

Despite being faster than the APRIORI-SD algorithm and the ability to handle medium size data sets, the DoubleBeam-SD algorithm is not able to handle large data sets due to space and time complexity. Also, DoubleBeam-RL cannot handle very large data sets. This is one of the main disadvantages of rule learning algorithms using a covering approach. Lower memory consumption could be achieved with more efficient data structures, while significant speedups could be gained with instance sampling and feature subset selection, as well as with parallelization of the algorithms. Due to the two beams, a degree of parallelization could be achieved with DoubleBeam algorithms.

Results from the multi-view clustering setting for determining groups of similar patients are underwhelming in terms of the quality of produced clusters. However, the results show the importance of autonomic symptoms for the quality of life of Parkinson's disease patients. Other approaches to multi-view learning should be explored as this could reveal interesting knowledge about the importance of symptoms and possibly new patterns of disease progression. An interesting direction for further work is also to explore other clustering approaches, in particular, hierarchical clustering. Attributes from the MDS-UPDRS and MoCA questionnaires can be ordered hierarchically and exploiting this characteristic may lead to better-defined groups of patients with similar symptoms. Transitions between such clusters could reveal more specific and detailed patterns of disease progression.

The rules describing the obtained clusters are either very general (merged view setting) or very specific (multi-view setting) and may not be of sufficient assistance to the clinicians. This is due to the nature of the used data, i.e. a vector of attribute sums (merged view) or a high-dimensional vector of attributes with numeric values. The performance of the methodology for disease progression patterns should be tested with only a handful of carefully chosen attributes. Current symptoms' evaluations are mainly numerical values which can include evaluation bias either by the clinicians or by the patients. Quantization of attribute evaluation by nominal values used in the clinicians' everyday practice (i.e. normal, non-problematic, problematic) could decrease the variance in attribute evaluation, potentially improving the performance of the methodology. An expert-assisted decrease of feature space dimensionality may contribute to more meaningful and helpful descriptions of groups of patients.

The discovered groups of patients with similar patterns of disease progression require further study and offer an opportunity for improved disease management. Investigation of the symptoms, mostly epidemiological, that are significant for the patients from each group can provide knowledge that can help the clinicians to prepare trajectories of disease progression for their patients and decide what steps they can take to slow the disease progression. An interesting route is to look also into the intersection of disease progression and medication therapy modification patterns described in Chapter 6. Knowing the characteristics of groups of patients that react well to therapy modifications and have a stable disease progression can rank therapies according to their success.

Descriptive models could be used to describe relevant subgroups of patients for whom certain therapy change patterns can have a positive influence on the control of symptoms, or therapy patterns which are more likely to lead to undesired side effects.

In time series analysis there are opportunities for analyzing more than one previous time point. Taking longer history into account might lead to the detection of groups of

patients which do not react well to the changes in antiparkinson medications.

In terms of therapy modifications, the issue of actual dosage changes in antiparkinson medications (*increase*, *decrease*, or *unchanged* medication dosage) should be addressed. One step closer towards a more personalized treatment of Parkinson's disease patients is to numerically predict dosage changes. Another interesting path for research is to see how the actual therapy influences the status of the patients. Currently, this thesis only addresses the Levodopa Equivalent Daily Dosage of antiparkinson medications and does not take into account any knowledge of how the frequency and prescribed dosage for one intake influence the improvement of the patients' status. We will also explore the impact of medication changes on the patients' status and their future symptoms.

Finally, the proposed technologies have to be integrated into the medical practice. We see an opportunity for this through recent initiatives for personalized medicine and exploitation of electronic health records. The methodology can be used as a foundation for building a decision support system, where the expert knowledge and the knowledge obtained from our methodology are combined in order to generate better decision support models.

Appendix A

A.1 Appendix to Chapter 5

Algorithm A.1 presents the pseudo code of the methodology for detection disease progression patterns using skip-grams. The groundwork for this methodology is presented in Chapter 5. The input to the methodology is the vector of assigned cluster labels (see Algorithm 1 in Chapter 5) and output is the vector of detected disease progression patterns. The output is generated over the cluster crossing sequences for all patients. More details can be found in Chapter 5.

Algorithm A.1: Pseudo code of the methodology for detection disease progression patterns using skip-grams. This methodology is presented in Chapter 5.

Input: : **c** — assigned cluster labels;
p — number of patients;
Output: : **diseaseProgressionPatterns** — patterns of disease progression;
Parameters: : s — number of skips;
n — lenght of n-gram;

```
// Initialize the vector of cluster crossing sequences to an empty list.
clusterCrossingSequences ← []

// Fill the vector for cluster crossing sequences with sequences from all patients.
for patient in [1 : p] do
  | clusterCrossingSequences ← getClusterCrossingSequence(c, p_i)
end

// Get patterns of disease progression using skip-grams.
patterns ← skipGrams(clusterCrossingSequences, s, n)
diseaseProgressionPatterns ← sort.ByDecreasingFrequency(patterns)
```

A.2 Appendix to Chapter 6

Algorithm A.2 presents the pseudo code of the methodology for detection of medications change patterns as a result of the current status of the patient. This is a multitask learning methodology that uses predictive clustering trees to detect dosage changes for all three groups of antiparkinson medications: levodopa, dopamine agonsists, and MAO-B inhibitors. The methodology is presented in Chapter 6.

A very important step in the methodology is the preparation of data. After the initial time-stamping of both the symptoms and medications data, the symptoms data is preprocessed, constructed, and evaluated according to the consulting clinicians' instructions. A short description of these instructions can be found in Chapter 6. Medication data is preprocessed in such a way that for each patient and each medication group, the algorithm

records dosage changes which occurred between two consecutive visits. These changes are associated with the *current*.

The symptoms are merged with the medication changes between the current and the next visit. Medication change patterns are detected using multitask predictive clustering trees (PCT).

Algorithm A.2: Pseudo code of the methodology for detection of medications change patterns from Chapter 6.

Input: : **S** — PPMI symptoms data;
M — PPMI medications data;
Output: : *PCT tree with medications change patterns;*

```
// Time-stamp symptoms data.  Associate the symptoms with the patient and the visit.
TS ← timeStampData(S)
// Time-stamp medications data.  Associate the medications data with the patient and the
   visit.
TM ← timeStampData(M)

// Pre-process symptoms data.  Select, construct, and evaluate symptoms according to the
   clinicians' instructions.
PTS ← preProcessSymptomsData(TS)
// Pre-process medications data.  For each instance (patient-visit pair) and each
   medications group record whether a dosage change has occurred between two consecutive
   time points (yes/no).
PTM ← timeStampData(TM)

// Merge pre-processed symptoms and medications data.
data ← mergeData(PTS, PTM)

// Detect medication change patterns using multitask predictive clustering trees (PCT).
   Target variables are the three antiparkinsonian medications groups:  levodopa, dopamine
   agonists, and MAO-B inhibitors.
PCT ← trainPCT(data, targetVariables = {levodopa, dopag, MAOB})
```

References

Adeli, E., Wu, G., Saghafi, B., An, L., Shi, F., & Shen, D. (2017). Kernel-based joint feature selection and max-margin classification for early diagnosis of Parkinson's disease. *Scientific Reports*, *7*(41069).

Adhikari, P. R., Vavpetič, A., Kralj, J., Lavrač, N., & Hollmén, J. (2014). Explaining mixture models through semantic pattern mining and banded matrix visualization. In *Proceedings of the 17th International Conference on Discovery Science, DS 2014* (pp. 1–12).

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB'94* (pp. 487–499).

Alberio, T., Bucci, E., Natale, M., Bonino, D., Di Giovanni, M., Bottacchi, E., & Fasano, M. (2013). Parkinson's disease plasma biomarkers: An automated literature analysis followed by experimental validation. *Journal of Proteomics*, *90*, 107–114.

Aliferis, C., Statnikov, A., Tsamardinos, I., Mani, S., & Koutsoukos, X. (2010a). Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, *11*(Jan), 171–234.

Aliferis, C., Statnikov, A., Tsamardinos, I., Mani, S., & Koutsoukos, X. (2010b). Local causal and Markov blanket induction for causal discovery and feature selection for classification part II: Analysis and extensions. *Journal of Machine Learning Research*, *11*(Jan), 235–284.

Aliferis, C., Tsamardinos, I., & Statnikov, A. (2003). HITON: A novel Markov blanket algorithm for optimal variable selection. In *Proceedings of the AMIA Annual Symposium* (Vol. 2003, pp. 21–25). American Medical Informatics Association.

Appice, A. (2017). Towards mining the organizational structure of a dynamic event scenario. *Journal of Intelligent Information Systems*, 1–29.

Appice, A., & Malerba, D. (2016). A co-training strategy for multiple view clustering in process mining. *IEEE Transactions Services Computing*, *9*(6), 832–845.

Arauzo-Azofra, A., Aznarte, J. L., & Benítez, J. M. (2011). Empirical study of feature selection methods based on individual feature evaluation for classification problems. *Expert Systems with Applications*, *38*(7), 8170–8177.

Atzmüller, M. (2015). Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *5*(1), 35–49.

Atzmüller, M., & Puppe, F. (2006). SD-Map - A fast algorithm for exhaustive subgroup discovery. In *Proceedings of Knowledge Discovery in Databases, PKDD 2006* (pp. 6–17).

Bazazeh, D., Shubair, R. M., & Malik, W. Q. (2016). Biomarker discovery and validation for Parkinson's disease: A machine learning approach. In *In Proceedings of the IEEE International Conference on Bio-engineering for Smart Technologies, BioSMART 2016* (pp. 1–6). IEEE.

Bega, D. (2017). Queen Square Brain Bank Diagnostic Criteria for Parkinson's Disease. http://bestpractice.bmj.com/topics/en-gb/147. Accessed: 2017/12/18.

Bence, J. R. (1995). Analysis of short time series: Correcting for autocorrelation. *Ecology*, *76*(2), 628–639.

Bezdek, J. C. (1981). *Pattern Recognition With Fuzzy Objective Function Algorithms*. Plenum Press, New York.

Bibal, A., & Frénay, B. (2016). Interpretability of machine learning models and representations: An introduction. In *Proceedings of the European Symposium on Artificial Neural Networks, ESANN 2016* (pp. 77–82).

Blockeel, H. (1998). *Top-down induction of first order logical decision trees* (Doctoral dissertation, Department of Computer Science, K.U.Leuven, Leuven, Belgium).

Blockeel, H., & De Raedt, L. (1998). Top-down induction of first-order logical decision trees. *Artificial Intelligence*, *101*(1-2), 285–297.

Blockeel, H., Raedt, L. D., & Ramon, J. (1998). Top-down induction of clustering trees. In *Proceedings of the 15th International Conference on Machine Learning, ICML 1998* (pp. 55–63).

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory, COLT 1998* (pp. 92–100). ACM.

Broder, A. Z., Glassman, S. C., Manasse, M. S., & Zweig, G. (1997). Syntactic clustering of the web. *Computer Networks and ISDN Systems*, *29*(8-13), 1157–1166.

Cai, X., Nie, F., & Huang, H. (2013). Multi-view k-means clustering on big data. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI 2013* (pp. 2598–2604).

Caruana, R. (1997). Multitask learning. *Machine Learning*, *28*(1), 41–75.

Caruana, R., Baluja, S., & Mitchell, T. (1996). Using the future to "sort out" the present: Rankprop and multitask learning for medical risk evaluation. In *Advances in Neural Information Processing Systems* (pp. 959–965).

Chaudhuri, K., Kakade, S. M., Livescu, K., & Sridharan, K. (2009). Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009* (pp. 129–136).

Choi, E., Schuetz, A., Stewart, W. F., & Sun, J. (2017). Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, *24*(2), 361–370.

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, *3*, 261–283.

Cleuziou, G., Exbrayat, M., Martin, L., & Sublemontier, J. (2009). CoFKM: A centralized method for multiple-view clustering. In *Proceedings of the 9th IEEE International Conference on Data Mining, ICDM 2009* (pp. 752–757).

Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning* (pp. 115–123).

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, *16*(3), 297–334.

Dalrymple-Alford, J., MacAskill, M., Nakas, C., Livingston, L., Graham, C., Crucian, G., ... Wells, S., et al. (2010). The MoCA: Well-suited screen for cognitive impairment in Parkinson disease. *Neurology*, *75*(19), 1717–1725.

De Alba, E., Mendoza, M. et al. (2007). Bayesian forecasting methods for short time series. *The International Journal of Applied Forecasting*, *8*, 41–44.

De Raedt, L., & Blockeel, H. (1997). Using logical decision trees for clustering. In *Proceedings of the IJCAI-97 Workshop on Frontiers of Inductive Logic Programming* (pp. 133–140). Springer.

De Vine, L., Zuccon, G., Koopman, B., Sitbon, L., & Bruza, P. (2014). Medical semantic similarity with a neural language model. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (pp. 1819–1822). ACM.

Ernst, J., & Bar-Joseph, Z. (2006). STEM: A tool for the analysis of short time series gene expression data. *BMC Bioinformatics*, *7*(1), 191.

Ernst, J., Nau, G. J., & Bar-Joseph, Z. (2005). Clustering short time series gene expression data. *Bioinformatics*, *21*(suppl 1), i159–i168.

Eskidere, Ö., Ertaş, F., & Hanilçi, C. (2012). A comparison of regression methods for remote tracking of Parkinson's disease progression. *Expert Systems with Applications*, *39*(5), 5523–5528.

European Parkinson's Disease Association. (2016). http://www.epda.eu.com/. Accessed: 2016/07/01.

Ferreira, J., Katzenschlager, R., Bloem, B., Bonuccelli, U., Burn, D., Deuschl, G., ... Kanovsky, P., et al. (2013). Summary of the recommendations of the EFNS/MDS-ES review on therapeutic management of Parkinson's disease. *European Journal of Neurology*, *20*(1), 5–15.

Fox, S. H., Katzenschlager, R., Lim, S.-Y., Ravina, B., Seppi, K., Coelho, M., ... Sampaio, C. (2011). The movement disorder society evidence-based medicine review update: Treatments for the motor symptoms of Parkinson's disease. *Movement Disorders*, *26*(S3), S2–S41.

Fürnkranz, J., Gamberger, D., & Lavrač, N. (2012). *Foundations of Rule Learning*. Springer.

Gamberger, D., & Lavrač, N. (2000). Confirmation rule sets. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, PKDD 2000* (pp. 34–43).

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

Gatsios, D., Rigas, G., Miljkovic, D., Seljak, B. K., & Bohanec, M. (2016). m-health platform for Parkinson's disease management. In *Proceedings of 18th International Conference on Biomedicine and Health Informatics, CBHI 2016*.

Gil, D., & Johnson, M. (2009). Diagnosing Parkinson by using artificial neural networks and support vector machines. *Global Journal of Computer Science and Technology*, *9*(4), 63–71.

Goebel, G., Seppi, K., Donnemiller, E., Warwitz, B., Wenning, G. K., Virgolini, I., ... Scherfler, C. (2011). A novel computer-assisted image analysis of [123 I] β-CIT SPECT images improves the diagnostic accuracy of parkinsonian disorders. *European Journal of Nuclear Medicine and Molecular Imaging*, *38*(4), 702–710.

Goetz, C., Luo, S., Wang, L., Tilley, B., LaPelle, N., & Stebbins, G. (2015). Handling missing values in the MDS-UPDRS. *Movement Disorders*, *30*(12), 1632–1638.

Goetz, C., Tilley, B., Shaftman, S., Stebbins, G., Fahn, S., Martinez-Martin, P., ... Zweig, R. (2008). Movement disorder society-sponsored revision of the unified Parkinson's disease rating scale (MDS-UPDRS): Scale presentation and clinimetric testing results. *Movement Disorders*, *23*(15), 2129–2170. doi:10.1002/mds.22340

Greene, D., Doyle, D., & Cunningham, P. (2010). Tracking the evolution of communities in dynamic social networks. In *Proceedings of Advances in Social Networks Analysis and Mining, ASONAM 2010* (pp. 176–183). IEEE.

Guthrie, D., Allison, B., Liu, W., Guthrie, L., & Wilks, Y. (2006). A closer look at skip-gram modelling. In *Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC 2006* (pp. 1–4).

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*(Mar), 1157–1182.

Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (2008). *Feature Extraction: Foundations and Applications*. Springer.

He, X., Kan, M.-Y., Xie, P., & Chen, X. (2014). Comment-based multi-view clustering of Web 2.0 items. In *Proceedings of the 23rd International Conference on World Wide Web* (pp. 771–782). ACM.

Hühn, J., & Hüllermeier, E. (2009). FURIA: An algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery*, *19*(3), 293–319.

Imhoff, M., Bauer, M., Gather, U., & Löhlein, D. (1998). *Time Series Analysis in Intensive Care Medicine*. SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund.

Kavšek, B., Lavrač, N., & Jovanoski, V. (2003). APRIORI-SD: Adapting association rule learning to subgroup discovery. In *Proceedings of 5th International Symposium on Intelligent Data Analysis* (pp. 230–241).

Kira, K., & Rendell, L. A. (1992). The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of AAAI 1992* (Vol. 2, pp. 129–134).

Klösgen, W. (1996). Explora: A multipattern and multistrategy discovery assistant. In *Proceedings of Advances in Knowledge Discovery and Data Mining* (pp. 249–271).

Kocev, D., Vens, C., Struyf, J., & Džeroski, S. (2007). Ensembles of multi-objective decision trees. In *Proceedings of the 18th European Conference on Machine Learning, ECML 2007* (pp. 624–631). Springer.

Kononenko, I. (1994). Estimating attributes: Analysis and extensions of Relief. In *Proceedings of the European Conference on Machine Learning* (pp. 171–182). Springer.

Kralj Novak, P., Lavrač, N., Zupan, B., & Gamberger, D. (2005). Experimental comparison of three subgroup discovery algorithms: Analysing brain ischemia data. In *Proceedings of the 8th International Multiconference Information Society* (pp. 220–223).

Kranjc, J., Podpečan, V., & Lavrač, N. (2012). ClowdFlows: A cloud based scientific workflow platform. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2012* (pp. 816–819).

Kumar, A., & Daumé, H. (2011). A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011* (pp. 393–400).

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

Lewis, S., Foltynie, T., Blackwell, A., Robbins, T., Owen, A., & Barker, R. (2005). Heterogeneity of Parkinson's disease in the early clinical stages using a data driven approach. *Journal of Neurology, Neurosurgery & Psychiatry*, *76*(3), 343–348.

Lin, J., Keogh, E., Wei, L., & Lonardi, S. (2007). Experiencing SAX: A novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, *15*(2), 107–144.

Liu, L., Wang, Q., Adeli, E., Zhang, L., Zhang, H., & Shen, D. (2018). Exploring diagnosis and imaging biomarkers of Parkinson's disease via iterative canonical correlation analysis based feature selection. *Computerized Medical Imaging and Graphics*, *67*, 21–29.

Liu, Y., Li, W., Tan, C., Liu, X., Wang, X., Gui, Y., ... Chen, L. (2014). Meta-analysis comparing deep brain stimulation of the globus pallidus and subthalamic nucleus to

treat advanced Parkinson disease: A review. *Journal of Neurosurgery*, *121*(3), 709–718.

Ma, L.-Y., Chan, P., Gu, Z.-Q., Li, F.-F., & Feng, T. (2015). Heterogeneity among patients with Parkinson's disease: Cluster analysis and genetic association. *Journal of the Neurological Sciences*, *351*(1), 41–45.

Marek, K., Jennings, D., Lasch, S., Siderowf, A., Tanner, C., Simuni, T., ... Chowdhury, S., et al. (2011). The Parkinson's Progression Markers Initiative (PPMI). *Progress in Neurobiology*, *95*(4), 629–635.

Michalski, R. S. (1969). On the quasi-minimal solution of the general covering problem. In *Proceedings of the 5th International Symposium on Information Processing, FCIP 1969* (Vol. A3, pp. 125–128).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mileva-Boshkoska, B., Miljkovic, D., Valmarska, A., Gatsios, D., Rigas, G., Konitsiotis, S., ... Bohanec, M. (2017). A state-transition decision support model for medication change of Parkinson's disease patients. In *Proceedings of the 20th International Multiconference Information Society, IS 2017* (pp. 63–66).

Miller, D. B., & O'Callaghan, J. P. (2015). Biomarkers of Parkinson's disease: Present and future. *Metabolism-Clinical and Experimental*, *64*(3), S40–S46.

Minnaert, B., Martens, D., De Backer, M., & Baesens, B. (2015). To tune or not to tune: Rule evaluation for metaheuristic-based sequential covering algorithms. *Data Mining and Knowledge Discovery*, *29*(1), 237–272.

Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence*, *18*(2), 203–226.

Murugesan, S., Bouchard, K., Chang, E., Dougherty, M., Hamann, B., & Weber, G. H. (2017). Multi-scale visual analysis of time-varying electrocorticography data via clustering of brain regions. *BMC Bioinformatics*, *18*(6), 236.

Napierala, K., & Stefanowski, J. (2015). Addressing imbalanced data with argument based rule learning. *Expert Systems with Applications*, *42*(24), 9468–9481.

National Collaborating Centre for Chronic Conditions. (2006). *Parkinson's disease: National Clinical Guideline for Diagnosis and Management in Primary and Secondary Care*. London: Royal College of Physicians.

Olanow, W., Watts, R., & Koller, W. (2001). An algorithm (decision tree) for the management of Parkinson's disease (2001): Treatment guidelines. *Neurology*, *56*(suppl 5), S1–S88.

Olesen, J., Gustavsson, A., Svensson, M., Wittchen, H.-U., & Jönsson, B. (2012). The economic cost of brain disorders in Europe. *European Journal of Neurology*, *19*(1), 155–162.

Parpinelli, R. S., Lopes, H. S., & Freitas, A. A. (2002). Data mining with an ant colony optimization algorithm. *IEEE Transactions on Evolutionary Computation*, *6*(4), 321–332.

Patel, S., Lorincz, K., Hughes, R., Huggins, N., Growdon, J., Standaert, D., ... Bonato, P. (2009). Monitoring motor fluctuations in patients with Parkinson's disease using wearable sensors. *IEEE Transactions on Information Technology in Biomedicine*, *13*(6), 864–873.

PD_manager: m-Health platform for Parkinson's disease management. (2015). EU Framework Programme for Research and Innovation Horizon 2020, Grant number 643706, 2015–2017. http://www.parkinson-manager.eu/.

Piccart, B., Struyf, J., & Blockeel, H. (2008). Empirical asymmetric selective transfer in multi-objective decision trees. In *Proceedings of the International Conference on Discovery Science, DS 2008* (pp. 64–75). Springer.

Pičulin, M., & Robnik-Šikonja, M. (2014). Handling numeric attributes with ant colony based classifier for medical decision making. *Expert Systems with Applications*, *41*(16), 7524–7535.

Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess end games. *Machine Learning. An Artificial Inteligence Approach*, *1*, 463–482.

Quinlan, J. R., & Cameron-Jones, R. M. (1993). FOIL: A midterm report. In *Proceedings of Machine Learning: European Conference on Machine Learning, ECML 1993* (pp. 3–20).

Ramani, R. G., & Sivagami, G. (2011). Parkinson disease classification using data mining algorithms. *International Journal of Computer Applications*, *32*(9), 17–22.

Rana, B., Juneja, A., Saxena, M., Gudwani, S., Kumaran, S. S., Behari, M., & Agrawal, R. (2015). Graph-theory-based spectral feature selection for computer aided diagnosis of Parkinson's disease using T1-weighted MRI. *International Journal of Imaging Systems and Technology*, *25*(3), 245–255.

Reijnders, J., Ehrt, U., Lousberg, R., Aarsland, D., & Leentjens, A. (2009). The association between motor subtypes and psychopathology in Parkinson's disease. *Parkinsonism & Related Disorders*, *15*(5), 379–382.

Reyes, O., Morell, C., & Ventura, S. (2015). Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context. *Neurocomputing*, *161*, 168–182.

Riviere, C. N., Reich, S. G., & Thakor, N. V. (1997). Adaptive Fourier modeling for quantification of tremor. *Journal of Neuroscience Methods*, *74*(1), 77–87.

Robnik-Šikonja, M., & Kononenko, I. (2003). Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, *53*(1-2), 23–69.

Ruz, G. A. (2016). Improving the performance of inductive learning classifiers through the presentation order of the training patterns. *Expert Systems with Applications*, *58*, 1–9.

Schieb, L. J., Mobley, L. R., George, M., & Casper, M. (2013). Tracking stroke hospitalization clusters over time and associations with county-level socioeconomic and healthcare characteristics. *Stroke*, *44*(1), 146–152.

Seppi, K., Weintraub, D., Coelho, M., Perez-Lloret, S., Fox, S. H., Katzenschlager, R. [Regina], ... Goetz, C., et al. (2011). The movement disorder society evidence-based medicine review update: Treatments for the non-motor symptoms of Parkinson's disease. *Movement Disorders*, *26*(S3).

Shang, R., Wang, W., Stolkin, R., & Jiao, L. (2016). Subspace learning-based graph regularized feature selection. *Knowledge-Based Systems*, *112*, 152–165.

Singh, G., & Samavedham, L. (2015). Unsupervised learning based feature extraction for differential diagnosis of neurodegenerative diseases: A case study on early-stage diagnosis of Parkinson disease. *Journal of Neuroscience Methods*, *256*, 30–40.

Stecher, J., Janssen, F., & Fürnkranz, J. (2014). Separating rule refinement and rule selection heuristics in inductive rule learning. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2014* (pp. 114–129).

Sun, X., Liu, Y., Li, J., Zhu, J., Liu, X., & Chen, H. (2012). Using cooperative game theory to optimize the feature selection problem. *Neurocomputing*, *97*, 86–93.

Szymański, A., Kubis, A., & Przybyszewski, A. W. (2015). Data mining and neural network simulations can help to improve deep brain stimulation effects in Parkinson's disease. *Computer Science*, *16*(2), 199.

Timmer, J., Gantert, C., Deuschl, G., & Honerkamp, J. (1993). Characteristics of hand tremor time series. *Biological Cybernetics*, *70*(1), 75–80.

Tsanas, A. (2012). *Accurate telemonitoring of Parkinson's disease symptom severity using nonlinear speech signal processing and statistical machine learning* (Doctoral dissertation, Oxford University, UK).

Tsanas, A., Little, M. A., McSharry, P. E., & Ramig, L. O. (2010). Enhanced classical dysphonia measures and sparse regression for telemonitoring of Parkinson's disease progression. In *In Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP 2010* (pp. 594–597). IEEE.

Tsanas, A., Little, M. A., McSharry, P. E., Spielman, J., & Ramig, L. O. (2012). Novel speech signal processing algorithms for high-accuracy classification of Parkinson's disease. *IEEE Transactions on Biomedical Engineering*, *59*(5), 1264–1271.

Tsanas, A., Little, M., McSharry, P., & Ramig, L. (2010). Accurate telemonitoring of Parkinson's disease progression by noninvasive speech tests. *IEEE Transactions on Biomedical Engineering*, *57*(4), 884–893.

Tzortzis, G., & Likas, A. (2009). Convex mixture models for multi-view clustering. In *Proceedings of the 19th International Conference Artificial Neural Networks, ICANN 2009* (pp. 205–214).

Vavpetič, A., Kralj Novak, P., Grčar, M., Mozetič, I., & Lavrač, N. (2013). Semantic data mining of financial news articles. In *Proceedings of the 16th International Conference on Discovery Science, DS 2013* (pp. 294–307).

Visser, M., Marinus, J., Stiggelbout, A. M., & Van Hilten, J. J. (2004). Assessment of autonomic dysfunction in Parkinson's disease: The SCOPA-AUT. *Movement Disorders*, *19*(11), 1306–1312.

Washburn, R. A., Smith, K. W., Jette, A. M., & Janney, C. A. (1993). The physical activity scale for the elderly (PASE): Development and evaluation. *Journal of Clinical Epidemiology*, *46*(2), 153–162.

Weintraub, D., Mamikonyan, E., Papay, K., Shea, J. A., Xie, S. X., & Siderowf, A. (2012). Questionnaire for impulsive-compulsive disorders in Parkinson's disease–rating scale. *Movement Disorders*, *27*(2), 242–247.

White, M., Zhang, X., Schuurmans, D., & Yu, Y. (2012). Convex multi-view subspace learning. In *Proceedings of Advances in Neural Information Processing Systems* (pp. 1673–1681).

Wrobel, S. (1997). An algorithm for multi-relational discovery of subgroups. In *Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery, PKDD 1997* (pp. 78–87).

Xu, C., Tao, D., & Xu, C. (2013). A survey on multi-view learning. *Neural Computing and Applications*, *23*(7–8), 2031–2038.

Zeng, Q., Patel, J. M., & Page, D. (2014). QuickFOIL: Scalable inductive logic programming. *VLDB Endowment*, *8*(3), 197–208.

Zhang, D., Shen, D., & ADNI. (2012). Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in Alzheimer's disease. *NeuroImage*, *59*(2), 895–907.

Zhao, J., Papapetrou, P., Asker, L., & Boström, H. (2017). Learning from heterogeneous temporal data in electronic health records. *Journal of Biomedical Informatics*, *65*, 105–119.

Zhou, J., Liu, J., Narayan, V. A., Ye, J., & ADNI. (2013). Modeling disease progression via multi-task learning. *NeuroImage*, *78*, 233–248.

Bibliography

Publications Related to the Thesis

Journal Articles

Valmarska, A., Lavrač, N., Fürnkranz, J., & Robnik-Šikonja, M. (2017). Refinement and selection heuristics in subgroup discovery and classification rule learning. *Expert Systems with Applications*, *81*, 147–162. doi:10.1016/j.eswa.2017.03.041

Valmarska, A., Miljkovic, D., Lavrač, N., & Robnik-Šikonja, M. (2018). Analysis of medications change in Parkinson's disease progression data. *Journal of Intelligent Information Systems*. doi:10.1007/s10844-018-0502-y

Valmarska, A., Miljkovic, D., Konitsiotis, S., Gatsios, D., Lavrač, N., & Robnik-Šikonja, M. (2018). Symptoms and medications change patterns for Parkinson's disease patients stratification. *Artificial Intelligence in Medicine (accepted)*.

Conference Papers

Valmarska, A., Robnik-Šikonja, M., & Lavrač, N. (2015). Inverted heuristics in subgroup discovery. In *Proceedings of the 18th International Multiconference Information Society* (Vol. 178, pp. 41–44).

Valmarska, A., Miljkovic, D., Robnik-Šikonja, M., & Lavrač, N. (2016). Multi-view approach to Parkinson's disease quality of life data analysis. In *Proceedings of the International Workshop on New Frontiers in Mining Complex Patterns* (pp. 163–178). Springer.

Valmarska, A., Miljkovic, D., Konitsiotis, S., Gatsios, D., Lavrač, N., & Robnik-Šikonja, M. (2017). Combining multitask learning and short time series analysis in Parkinson's disease patients stratification. In *Proceedings of the Conference on Artificial Intelligence in Medicine in Europe* (pp. 116–125). Springer.

Biography

Anita Valmarska was born on June 12, 1987 in Berovo, Macedonia. She finished primary school in Berovo in 2002 and secondary education in Maribor in 2006. In 2006, she started her studies at the University of Ljubljana, Faculty of Computer and Information Science. She defended her BSc thesis entitled "Analysis of Citation Networks" under the supervision of Prof. Dr. Janez Demšar in 2014. During her secondary and university education she received scholarships for talented students awarded by the Macedonian and Slovenian Ministry of Education.

In 2014, she was employed as a junior researcher at the Jožef Stefan Institute, Slovenia, under the supervision of Prof. Dr. Nada Lavrač at the Department of Knowledge Technologies. In the same year, she started her doctoral studies in the PhD programme "Information and Communication Technologies" at the Jožef Stefan International Postgraduate School under the supervision of Prof. Dr. Marko Robnik-Šikonja and co-supervision of Prof. Dr. Nada Lavrač.

Her research is in the field of Parkinson's disease patients data analysis with data mining methods, and development of new methods for descriptive learning (algorithms for classification rule learning and subgroup discovery). In the field of Parkinson's disease patients data analysis, she works on new methods for disease progression analysis. She collaborated in the EU-funded project PD_manager, where she developed new analytical approaches for Parkinson's disease patient quality of life management. She attended several courses and summer/winter schools including: BigData 2015, the 2nd HBP Education Workshop on Future Medicine, and DeepLearn 2017. She presented her work in published journal papers and at international conferences and workshops.