Machine Learning on Data with Inconsistencies and its Granular Properties

Marko Palangetić

Dissertation submitted in fulfilment of the requirements for
the degree of Doctor of Science: Computer Science

Academic year 2021–2022

Department of Applied Mathematics,
Computer Science and Statistics
Faculty of Sciences
Ghent University

Supervisors

Prof. dr. Chris Cornelis
Department of Applied Mathematics, Computer Science and Statistics,
Ghent University, Belgium

Prof. dr. Roman Słowiński
Institute of Computing Science,
Poznań University of Technology, Poland
Systems Research Institute,
Polish Academy of Sciences

Prof. dr. Salvatore Greco
Department of Economics and Business,
University of Catania, Italy
Portsmouth Business School,
University of Porthsmuth, United Kingdom

Other members of the examination board

Prof. dr. Kris Coolsaet (chair)
Department of Applied Mathematics, Computer Science and Statistics,
Ghent University, Belgium

Prof. dr. Richard Jensen
Department of Computer Science,
University of Aberystwyth, United Kingdom

Prof. dr. Yvan Saeys
Department of Applied Mathematics, Computer Science and Statistics,
University of Ghent, Belgium

Dr. Sarah Vluymans
Data Scientist, VDAB, Belgium

Prof. dr. Willem Waegeman
Department of Data Analysis and Mathematical Modelling,
Ghent University, Belgium

This work was supported by
Research Foundation Flanders (grant no. G0H9118N)

Contents

List of Tables

List of Figures

Summary

In this thesis, we tackle the problem of inconsistency in a dataset represented by an information table, i.e., a finite set of data instances described by condition attributes (independent variables) and one decision attribute (dependent variable). The goal is to identify a causal relationship between the condition and decision attributes based on the observed set of instances. An information table is consistent if for every two instances that relate in a certain way on the condition attributes, they relate in a similar way on the decision attribute. For example, two instances that are indiscernible on conditions attributes should have the same decision. In the opposite case, we call the instances inconsistent.

Tackling the problem of inconsistency means to remove inconsistencies from a dataset by changing values of the decision attribute in order to obtain a consistent dataset. This problem is approached from multiple perspectives. First, we recall the traditional rough set approach and its variations. We discuss how to improve these approaches, make them more robust, modelling graduality in the information through the use of fuzzy logic, and perform experiments that empirically test their robustness. We also discuss their granular properties, i.e., the ability to represent rough set approximations as unions of simple, meaningful sets called granules. We explain how the granules can be interpreted as decision rules and therefore be used in rule induction methods. We also analyze the granularity properties of the previously discussed robust versions of rough sets.

On top of the hybrid fuzzy-rough approach, we consider the inconsistency problem from the statistical learning perspective. Here, the assumption is that the instances in an information table are realizations of random variables, or, more precisely, of a fuzzy random variable. We solve the inconsistency problem as an optimization problem, i.e., we remove inconsistencies by minimizing the loss of information. We also discuss granular properties of this approach, i.e., we study if the pro-

posed approach has a potential to be used in rule induction methods.

At the end, we develop an instance-based classification procedure based on the proposed statistical approach to inconsistency handling. We compare its performance with other similar machine learning classifiers and we stress its biggest strength: interpretability, i.e., the ability to clearly explain the predicted classification of new instances by similarity to instances from the original dataset.

Nederlandstalige samenvatting

In dit proefschrift behandelen we het probleem van inconsistentie in een dataset die wordt voorgesteld door een informatietabel, d.w.z., een eindige verzameling data-instanties die worden beschreven door conditionele attributen (onafhankelijke veranderlijken) en één beslissingsattribuut (afhankelijke veranderlijke). Het doel is het identificeren van een causaal verband tussen de conditionele attributen en het beslissingsattribuut gebaseerd op de geobserveerde verzameling instanties. Een informatietabel is consistent als voor elke twee instanties die op een bepaalde manier met elkaar in verband staan voor de conditionele attributen, op gelijkaardige wijze kunnen gerelateerd worden voor het beslissingsattribuut. Bijvoorbeeld, twee instanties die ononderscheidbaar zijn voor de conditionele attributen moeten dezelfde beslissing hebben. In het andere geval noemen we de instanties inconsistent.

Onze aanpak van het consistentieprobleem komt neer op het verwijderen van inconsistenties uit een dataset door de waarden van het beslissingsattribuut te wijzigen om een consistente dataset te bekomen. We benaderen dit probleem vanuit verschillende invalshoeken. Eerst brengen we de traditionele ruwverzamelingenaanpak en diens variaties in herinnering. We bespreken hoe we deze aanpakken kunnen verbeteren, maken ze robuust, waarbij we gradualiteit in de informatie modelleren aan de hand van vaaglogica, en we voeren experimenten uit die hun robuustheid empirisch evalueren. We bespreken eveneens hun granulaire eigenschappen, met andere woorden, we onderzoeken de mogelijkheid om de benaderingen uit de ruwverzamelingenleer voor te stellen als unies van eenvoudige, betekenisvolle verzamelingen die we granules noemen. We leggen uit hoe de granules geïnterpreteerd kunnen worden als regels en derhalve gebruikt kunnen worden in regelinductiemethoden. We analyseren eveneens de granulariteitseigenschap-

pen van de eerder besproken robuuste versies van ruwverzamelingen.

Bovenop de hybride vaagruwe aanpak bestuderen we het inconsistentieprobleem ook vanuit het perspectief van statistisch leren. De aanname hierbij is dat de instanties in een informatietabel uitkomsten zijn van toevalsveranderlijken, of, meer precies, van een vaagtoevalsveranderlijke. We lossen het inconsistentieprobleem op door middel van optimalisatieproblemen, d.w.z., we verwijderen inconsistenties door het minimaliseren van het informatieverlies. We bespreken ook de granulaire eigenschappen van de aanpak, d.w.z., we bestuderen of de voorgestelde aanpak potentieel bezit om gebruikt te worden in regelinductiemethoden.

Tot slot ontwikkelen we een instantiegebaseerde classificatiemethode gebaseerd op de voorgestelde statistische aanpak van inconsistentie. We vergelijken de performantie ervan met andere gelijkaardige machine learning classifiers en we belichten de grootste troef van onze methode: transparantie, d.w.z., de mogelijkheid om de voorspelde classificatie van nieuwe instanties uit te leggen a.d.h.v. de gelijkenis met instanties uit de originele dataset.

Acknowledgements

First of all, I want to express my sincere gratitude to my first advisor, Prof. Dr. Chris Cornelis for his immense help with everything related to my PhD. He was always available for my questions and requests, ready to maximally assist up to his abilities, even outside of working hours.

I would also like to extend my deepest gratitude to my other advisors, Prof. Dr. Roman Słowiński and Prof. Dr. Salvatore Greco for their help and inspiring discussions. Their wise words and recommendations are integral parts of this thesis.

I want to thank the other members of the examination board: Prof. Dr. Kris Coolsaet, Prof. dr. Richard Jensen, Prof. dr. Yvan Saeys, Dr. Sarah Vluymans and Prof. Dr. Willem Waegeman for their useful and constructive feedback during the private defence. A special thank goes to Dr. Sarah Vluymans whose feedback improved the thesis significantly.

I would like to thank Oliver and Olha, the members of our research group, for interesting and at times humorous discussions on our research topics.

I am thankful for spending 4 years at the Department of Applied Mathematics, Computer Science and Statistics which is a very welcoming and friendly research environment. I was lucky to be surrounded by many smart and pleasant colleagues. I would like to thank Kelly, Oliver Dukes, Hans, Christophe, Hege, Josephina, Jeroen, Milan, Thang, Elke, Koen, Vahe, Paula, Louis, Ludger and Johan for our lunches, dinners and beer sessions. A special mention goes to Paweł, Olha, Viacheslav, Vitalii, Oliver Lenz and Camila with whom I spent many days having coffees, playing board games, cooking dinners and in general, having a very nice time.

Many thanks go to our department's administration: Ann, Hilde, Wouter and Herbert for taking care of my day-to-day issues.

Iskoristio bih priliku da iskažem zahvalnost mojim roditeljima, ocu

Miši i majci Vukosavi za nesebičnu podršku tokom cjelokupnog školovanja.

Na kraju, posebno se želim zahvaliti Slađani na pokazanom razumijevanju i korisnim savjetima koje mi je davala.

Chapter 1

Introduction

We start this introductory chapter with a discussion in Section 1.1 on the topics of machine learning and supervised learning problems. Then, we introduce the concept of data with inconsistencies, which is related to supervised learning, and we briefly recall how inconsistencies are handled in rough set theory. We also position rough sets within the broader framework of granular computing. Next, in Section 1.2, we enlarge our research perspective to the realm of fuzziness: we motivate the use of fuzzy relations, fuzzy membership degrees and fuzzy granules in data analysis, and recall the fuzzy rough set approach and its robust extensions for handling graded inconsistencies. In Section 1.3, we recall the statistical learning approach to inconsistency correction w.r.t. a crisp preorder relation, which constitutes the basis for the development of methods for handling inconsistency w.r.t. a fuzzy relation. In Section 1.4, we discuss interpretability methods used in machine learning, and how methods for inconsistency correction can be used to develop interpretable machine learning models. Finally, in Section 1.5, we provide an overview of the different chapters of this thesis.

1.1 Data inconsistency in machine learning and the rough set approach

1.1.1 Introduction to machine learning

Machine learning (ML) is a subfield of artificial intelligence where one tries to train machines to learn from the available previous experience. This experience is expressed through data. Such learning from data

leads to a model that is able to make predictions or decisions without being explicitly programmed to do so [90]. The area of machine learning can be roughly divided into three main subfields:

- **Supervised learning**: here, the computer is presented with a set of desired input-output or condition-decision pairs and the aim is to learn a general function that will obtain a decision for the provided conditions. Such a function should be able to properly map new, unseen input conditions into correct decisions. Measuring how close the function output is to the correct decisions on new, unseen data is the main indicator of quality of a supervised learning method.

- **Unsupervised learning**: in this case, the computer is presented with a set of instances without specific labels (decisions). The goal is to learn the hidden patterns and structures that the data exhibit. Usually, there is no exact way to evaluate the quality of unsupervised learning methods and the quality usually depends on individual perception.

- **Reinforcement learning**: in this subfield of machine learning, the agent interacts with a dynamical environment in which it must reach a certain goal. The agent receives feedback from the interactions and using this feedback, it adjusts its actions for the future in order to accomplish the goal. The feedback is in the form of rewards, and the actions are adjusted in order to maximize the future rewards.

Apart from the three main subfields, there are some variations such as semi-supervised learning, self-supervised learning, etc. In this thesis, we focus on supervised learning, and more precisely on the prediction problem, where we want to find a proper mapping from the condition space to the decision space.

Data in the condition space can be represented using various forms: tabular, images, text, etc. In the tabular form, every instance in the condition space is represented with a vector of numerical or nominal values. The description of each entry in such a vector is called a condition attribute of the corresponding instance. All such instances, together with their vector representation, form a table which is called an information table.

The decision values can come from different spaces. If the values are nominal, i.e., they come from a finite discrete space, the prediction prob-

lem is called a classification problem. If the values are real numbers, we deal with a regression problem. These are the two main types of prediction problems. The decision space can also consist of images, text or some other complex objects. In such case, we deal with structured prediction. The decision attribute, together with the condition attributes from an information table forms a decision table.

Various algorithms are used to construct such a mapping and all of them rely on different assumptions. For example, linear models, which include linear regression and logistic regression, rely on the assumption that the decision is obtained as a linear combination of the condition attributes, while classification and regression trees (CART) assume that with a hierarchical set of rules one can describe the influence of condition attributes on the decision attribute [47, 70]. Other methods that are widely used in supervised learning are instance-based methods (k-nearest neighbors (kNN) [46], locally weighted regression (LWR) [119], isotonic regression [12], ...), kernel-based methods (support vector machine (SVM) [3, 29], kernel ridge regression [100], ...), sequential rule-based methods (RIPPER [23], LEM2 [66], FURIA [74], ...), ensemble methods (random forests [71], AdaBoost [48], ...), etc.

In the last decade, the most popular models for supervised learning have been those based on artificial neural networks (ANNs) [116, 120]. Such structures, motivated by a simplified analogy to the human brain, outperform all other methods in many areas of supervised learning, especially in computer vision (analysis of images) and natural language processing (analysis of text). Their flexibility and the possibility to arbitrarily increase the number of parameters, enable them to successfully mimic human capabilities in certain tasks like face recognition, machine translation and speech recognition. However, they require significantly larger amounts of data, their training is computationally expensive and may require specialized equipment like graphical processing units (GPUs) or tensor processing units (TPUs) [81, 91, 101].

In this thesis, we want to explore the assumption of consistency between the condition attributes and the decision attribute. The property of consistency is introduced in the following subsection.

1.1.2 The assumption of consistency

Assume we have a prediction problem where we wish to assign a decision label to a given instance described by a number of condition attributes. A case at hand may be the patient records of a hospital, where

patients (instances) are described by their medical parameters (condition attributes), and the decision attribute refers to a patient's diagnosis for a given disease. If two patients exhibit identical medical conditions, we expect their diagnoses to be the same as well.

In general, we say that two instances are consistent w.r.t. a given relation, if their relation on the condition attributes implies the same type of relation on the decision attribute. In the opposite case, the instances are said to be inconsistent. We say that a single instance is consistent if it is consistent with all other instances as pairs. In the example given above, the considered relation is indiscernibility, which measures if two instances (patient records in this case) are identical. However, as we will see throughout the thesis, other types of relations also arise naturally.

In practice, due to perturbation in data caused by incomplete knowledge or by random effects that occur during data generation, datasets contain instances that are not consistent. The presence of inconsistent instances obviously creates obstacles for machine learning algorithms that try to extract meaningful patterns from data. To make a dataset consistent, different approaches have been taken.

1.1.3 Consistency in monotone classification problems

An example of inconsistency that is assumed in ordinal classification problems is the one w.r.t. monotonicity constraints.

Ordinal classification (also called ordinal regression) problems constitute a very important part of machine learning and statistical analysis [69]. In ordinal classification, the goal is to predict for a certain instance u from set U, described by its values for a set of condition attributes, one of K different ordinal class labels $y \in \{1, \ldots, K\}$.

Ordinal classification problems exploit the existing ordering on the decision attribute. In some cases, an ordering also exists on the condition attributes. One way to incorporate this knowledge is through so-called monotonicity constraints. For a given preorder (dominance) relation on the set of instances U based on the condition attributes, the monotonicity constraints can be formulated as follows [56]: if instance u_1 dominates u_2 w.r.t. the given dominance relation on the condition attributes, then u_1 should be assigned to the same or to a better decision class than u_2. In such a case, we may also say that u_1 and u_2 are consistent w.r.t. the given dominance relation. Monotonicity constraints are intuitively desirable; for example, if we have two companies where one of them has better financial parameters, then it should also have a lower bankruptcy

risk than the other.

Ordinal classification problems that include monotonicity constraints are called monotone classification problems. They arise in many areas, including medical diagnosis [22], survey data [20], estimation of bankruptcy risk [62], house pricing [106] and others. A comprehensive survey of monotone classification methods is given in [19].

1.1.4 Rough sets

One of the first methodologies developed to handle inconsistency with respect to indiscernibility is the rough set approach, introduced by Pawlak [103] in 1982. Given a decision table where instances are described by a set of attributes, Pawlak's approach produces two sets, called lower and upper approximation. They represent elements being, respectively, necessarily consistent (lower approximation), and possibly consistent (upper approximation) with knowledge contained in the decision table. The original theory was designed to exploit only nominal information carried by attributes, and relies on an equivalence relation, expressing indiscernibility or equality between elements.

Greco et al. [56] extended this framework with their Dominance-based Rough Set Approach (DRSA) allowing attributes to have ordinal value sets, and replacing the indiscernibility relation with a dominance relation. To distinguish between the two approaches, Pawlak's original theory is also called Indiscernibility-based Rough Set Approach (IRSA).

1.1.5 Granular computing

Granular computing is a paradigm in information processing which includes a segmentation of complex information into smaller pieces called information granules; it has been applied to diverse models in data analysis [11, 104, 137].

An information granule (or simply a *granule*) is a collection of instances that can be interpreted jointly. For example, an image of a human body can be segmented into certain body parts that have precise meanings. Also, those parts can later be segmented into even smaller meaningful parts, etc. The previous example shows the hierarchical nature of granulation, i.e., the definition of granules depends on the level of detail that we want to capture. Granules are usually constructed based on a common association (indiscernibility, similarity, functionality, proximity, coherency, etc.) of instances [36, 140].

Both IRSA and DRSA relate to granular computing, as they possess a so-called granular representation; indeed, lower and upper approximations can be represented as unions of simple sets or granules, induced from the data [138].

The granular representation of rough sets is particularly useful from the perspective of rule induction. The problem of rule induction for classification tasks amounts to generating a set of decision rules which relate descriptions of instances by subsets of attributes with particular decision classes. Basic granules, from which rough sets are composed, can be interpreted as human readable "*if...*, *then...*" rules, and can be used to construct a rule-based inference system as a prediction model. Well-known examples of rule induction algorithms are the LEM2 algorithm [66] and the MODLEM algorithm [67] for IRSA, and the DomLEM algorithm [60] for DRSA.

1.2 Fuzzy set theory and fuzzy rough sets

1.2.1 Fuzzy sets in supervised learning

Fuzzy logic and fuzzy set theory, introduced by Zadeh [144] in 1965, are used to model partial truth of logical expressions. In other words, the expression is not necessarily true or false, but it possesses a degree of truth represented by a value from the interval $[0, 1]$. Value 0 stands for a completely false statement, while value 1 represents a completely true statement.

Two ways to utilize fuzzy logic in data analysis are:

- fuzzy relations,
- fuzzy membership degrees in decision classes.

Fuzzy relations are able to model relationships between instances represented with attribute vectors. Namely, the usual crisp relations may distinguish only between two extreme cases: either instances relate or do not. Fuzzy relations, on the other hand, can express a degree to which two instances relate on a scale between 0 and 1. This is suitable for example to model gradual similarity or dominance between vector representations of instances or different structural representations (graphs, strings, DNA chains, ...).

Fuzzy membership degrees can relax the belonging to a particular decision class in classification problems by assuming that an instance

can belong to multiple decision classes by different degrees. Usually in classification problems, we assume that instances belong to a single (or more in multi-label classification) decision class. However, in some cases it is more realistic to assume that they belong to a decision class by some degree which brings the situation where one instance belongs more to a decision class than another instances. An example for this can be found in recommender systems.

A degree of preference for a certain product by a user can be modeled using values between 0 and 1. It is also possible that collected data contain only binary preferences while the underlying degree of preference is hidden and can be estimated using machine learning techniques. For example, when we use a movie streaming service, we are often asked to rate a movie with either "like" or "dislike". We have only two options. But in reality, preference of movies is gradual; we like some movies more than others, but this graduality cannot be expressed with only two options: like and dislike.

Another application of membership degrees can be found in sentiment analysis problems, where we want to detect the presence of certain emotions in text. Here, we can also assume that emotions like hate or joy are present in various degrees in different texts. But very often, we have the situation as in the previous example where the collected decision labels are binary and where those decision labels are dependent on the individual perception of the person that was labeling the text, which produces an additional perturbation. Therefore, modeling the presence of emotions with fuzzy membership degrees is suitable in these situations. We will consider a related didactic example in Chapter 8.

1.2.2 Fuzzy rough set theory

The integration of fuzzy logic and IRSA was initially proposed by Dubois and Prade [41], allowing to define the rough approximations on decision values represented with fuzzy membership degrees, using a fuzzy indiscernibility relation. A similar extension of DRSA to fuzzy set theory was proposed by Greco et al. [55].

In a similar way as rough sets produce lower and upper approximations as sets without inconsistencies, fuzzy rough sets produce fuzzy lower and upper approximations that do not posses inconsistencies w.r.t. a fuzzy relation. The ability to use fuzzy relations that are suitable for modeling similarity between instances with numerical attributes, placed fuzzy rough sets as an interesting research topic in the machine learn-

ing community. Apart from being used in usual classification tasks (e.g. Fuzzy Rough Nearest Neighbour model [78]), Fuzzy IRSA has also been applied extensively in other machine learning specific domains like fuzzy rule induction methods [149, 148], instance-based models [78, 14], attribute selection [26, 108, 130, 131, 142, 143], instance selection [77, 93], imbalanced classification [111], multi-label classification [126], and so on.

It is well-known that the classical definitions of fuzzy rough sets in both the indiscernibility and dominance case are vulnerable to possible perturbations, in a similar way as their crisp counterparts: small fluctuations in data may cause huge changes in membership values of the approximations. For this reason, various robust versions of the fuzzy rough approximations were proposed [25, 27, 43, 97]. In this thesis, we will focus on the Ordered Weighted Average (OWA) approach. OWA has been shown to improve fuzzy IRSA in handling outliers and noisy data [28, 111, 127, 125, 128]. Also, it was shown that the OWA extension provides the best trade-off between theoretical properties and experimental performance among robust models [35].

In contrast to crisp sets, the granular properties of fuzzy rough sets do not stem directly from the proposed definitions. Degang et al. [36] were the first to show that fuzzy IRSA indeed has a granular representation, which means that fuzzy rough approximations can be represented as a union of simple fuzzy sets or fuzzy granules. Later, Yao et al. [139] pointed out that the symmetry of the fuzzy relation is not essential for the granular representation, and hence it can be extended to fuzzy DRSA as well.

The granularity of fuzzy rough sets is also useful for rule induction. In this case, we obtain a fuzzy inference system, with flexible fuzzy rules instead of strict ones [79, 149]. The main advantage of fuzzy rules is that they can model complex patterns of data, and still keep an intuitive interpretation of these patterns.

1.2.3 Fuzzy sets and granular computing

Zadeh [146] identified granulation as one of three basic concepts in underlying human cognition, the other two being organization and causation. While organization represents the integration of parts into a whole, granulation refers to the opposite process. With fuzziness as a key part of the granulation in human cognition, humans are able to make reasonable decisions in a world that is characterized with partial knowledge,

partial certainty, partial truth and imprecision in general.

With the help of fuzzy logic, one can introduce the concept of a fuzzy granule [145] where every instance has a degree of membership to a certain granule. Fuzzy granules are useful when it is hard to determine sharp boundaries of pieces obtained from a segmentation of a complex object. In such case, soft boundaries are expressed using fuzzy sets.

In this thesis, fuzzy granules are identified in decision tables based on the concept of data consistency defined in Section 1.1. For a consistent instance (w.r.t. a given fuzzy relation), a fuzzy granule is formed as a conjunction of two concepts:

- the fuzzy set of instances that relate to the given consistent instance, and
- the association of the consistent instance to a particular decision.

Due to consistency, the instances that relate (w.r.t. a given fuzzy relation, like, e.g. indiscernibility or dominance) to a given consistent instance will be associated to the same decision or to a decision that relates to the decision of the consistent instance. In a classification problem, we have decision classes and the association of the consistent instance refers to the membership of the instance to a decision class. In a regression problem, the association refers to the numerical value that the consistent instance takes in the decision attribute.

1.3 Kotłowski-Słowiński approach

The rough set methods are considered to be extreme in their way of inconsistency handling. Namely, the resulting approximations assign all inconsistent instances to one decision class, either to the approximated decision class (upper approximation case) or to the opposite class (lower approximation case). This leads to a situation where the approximations may be significantly different from the original decision classes. Then, the question arises if the inconsistency can be corrected with the least possible amount of change of decision values. This can be achieved if handling inconsistency is formulated as an optimization problem. For this purpose, the statistical learning perspective of inconsistency correction in monotone classification problems was considered by Kotłowski and Słowiński [89]. They provided statistical foundations of the monotonicity constraints and developed a machine learning method to incorporate them into data analysis. In particular, they designed an optimization procedure that removes inconsistencies in data ("monotonizes"

them) at the least possible cost w.r.t. a certain loss function (used to measure the difference between the original decision labels and the new decision labels). It produces a new set of labels called a monotone approximation. The authors also showed that, for a specific family of loss functions called *monotone loss functions*, the optimization problem can be solved efficiently using linear programming. Finally, the relabeled sets possess the property of granular representation, meaning that they can be represented as unions of meaningful granules. This approach also generalizes standard rough sets and provides another probabilistic view of them. The approach found its application in the same areas as DRSA [88], as well as in the development of rule induction and ensemble rules methods [37]. A well-known representative of this approach is the widely used isotonic regression model [12] which uses squared error loss as its loss function. In the remainder of the thesis, we refer to this method as the *KS approach*.

1.4 Interpretability of machine learning models

1.4.1 Motivation for the interpretability

Many machine learning models have strong prediction performance and they provide great results in practice. A well-known example are artificial neural networks, the main tool used in computer vision, natural language processing, sentiment analysis and many other fields. However, neural networks are complex structures by their nature, and we do not know too much about how they learn from data. Sometimes this is not important (e.g., it is not necessary to know how the face identification on a smartphone works) but there are cases where the knowledge about a model may have a significant impact.

The first case is when we are interested in knowledge generated by a machine learning algorithm. High prediction performance means that the algorithm learned a lot about the relationship between data, detected some trends, found impact of different factors on the output, etc. So, we would like to extract all that important information from the algorithm for which we need reliable interpretation methods. For example, if a client is rejected for a loan in a bank based on a ML algorithm, they would like to know why that happened.

The second case may be for the purpose of debugging an ML algorithm. Apart from valuable knowledge, the algorithm may also learn

things which do not correspond to the reality of the problem [113]. This happens mainly due to poor data quality or due to the lack of robustness of the model to noise or outliers. We want to avoid making predictions based on false knowledge, even if prediction performance is high at first impression. To debug the algorithm, we need to "unpack" it, to see what it learned and to fix the possible issues.

The third case is when a wrong prediction may have a significant impact. An example are self-driving cars. It was shown that systems for recognizing objects on a road that are based on deep learning, are vulnerable to adversaries [42]. In particular, the authors developed very simple single-color stickers and positioned them on a **STOP** sign. While such stickers would not affect an ordinary human in recognizing the sign, a self-driving car recognized it as a speed limit. This could lead to fatalities. Therefore, users need to know how ML algorithms used in such sensitive environment work, to be able to avoid negative consequences.

1.4.2 Interpretable models

The following part is loosely motivated by [98]. The interpretation methodology can be roughly divided into two families: models that are interpretable by their construction and methods that are used to interpret models with low interpretability, like ANNs (or black-box models, as they are sometimes called).

Interpretable models are those whose prediction making process is understandable by humans to some extent. We identify two types of interpretability here:

- *modular interpretability*, where building units, like parameters, can be interpreted jointly, and
- *local interpretability*, where making individual predictions can be comprehended by a human.

The models that are considered as interpretable are linear models, which include linear and logistic regression [112], and rule-based models like CART [109] or rough set-based rule induction [66], while the models interpretable to some extent are instance-based models like kNN [46] or fuzzy rough set based k-Fuzzy Rough Nearest Neighbour (kFRNN) [78, 110] and prototype-based models like Learning Vector Quantization (LVQ) [87]. The latter two groups are interpretable "to some extent", because there are versions of them that are not interpretable. This happens for example if k is too large for the instance-based methods, or

when the learned prototypes are meaningless for the prototype-based methods. More information about interpretability of these methods is given in Section 8.4.

The linear models are interpretable on a modular level, i.e., the linear coefficients have a specific interpretation. The rule-based methods can belong to both types of interpretable methods. They are definitely locally interpretable since for every prediction, a particular rule can be identified that leads to that prediction. The decision tree structure of CARTs can be seen as modularly interpretable, since all splits during the training, together with the hierarchical structure, can be understandable. Prototype-based methods also belong to both types of interpretability. The local interpretability holds from the fact that for every new prediction, a prototype that contributed to the prediction can be identified. Also, the set of prototypes, as a representative of the data distribution, can be seen as a form of modular interpretability if the prototypes are interpretable. In all mentioned cases, the interpretability is also dependent on the size of the structure, i.e., the number of parameters (or attributes) in linear models, the number of splits for CART (or the number of rules in general) and the number of prototypes. Smaller structures will lead to a better understanding and therefore some size constraints are welcome in this case. Finally, instance-based methods are purely locally interpretable because they are inherently local as such. More technical details on the aforementioned locally interpretable methods is given in Sections 8.1 and 8.4.

1.4.3 Methods for interpreting black-box models

Methods that are used to explain black-box models are based on the idea that separating explanations from modeling can lead to better interpretability [114]. Such methods are called model-agnostic. Their biggest advantage is their flexibility, i.e., a user can use any ML model, since explanations are independent from them. This family of methods consists of two different subfamilies: global model-agnostic methods and local model-agnostic methods. Global model-agnostic methods tend to explain methods as a whole, while the local counterpart concentrates mainly on individual predictions.

Global model-agnostic methods include various procedures for measuring the contribution of individual attributes like partial dependence plots [49, 64], accumulated local effects [8], permutation feature importance [18, 45], etc. All these methods want to assess how individual

attributes are important to the model as a whole.

Another type of global model-agnostic methods are global surrogate methods. These are interpretable methods that are applied to predictions of a black-box method [4, 82]. For example, if an ANN is applied to a dataset, then the decision attribute values are changed based on the predictions of the model and then, an interpretable model, such as a linear or rule-based model, is applied using the new decision values. In this way, one wants to better understand the black-box model using the interpretable one.

The third type of global model-agnostic methods are prototype-based methods. We already mentioned a prototype-based interpretable models like LVQ. While in the case of interpretable models the prototypes are not necessarily instances from the dataset, here they come from the dataset, which can give better explanations under the assumption that the instances are meaningful. A representative method, based on maximum mean discrepancy (MMD) [65], is called MMD-critic and identifies instances that are representatives of the data distribution (prototypes) as well as instances that do not follow the distribution determined by prototypes (critics). This is a representative of instance-based methods for interpretation that will be discussed later.

Local model-agnostic methods are concentrated on explaining individual predictions of black-box models rather than the model as a whole. We discuss three representative methods from this group: Local interpretable model-agnostic explanations (LIME), Shapley values and counterfactuals.

LIME methods work under the assumption that a very complex model can be simple in the neighbourhood of the instance for which we want to explain the prediction [113]. Therefore, learning an interpretable model in the neighbourhood can explain what drives the prediction of the instance. The interpretable models which are used in this case are either linear [113] or rule-based [115].

Shapley values, motivated from game theory [122], is a method that aims to explain how individual attributes contribute to the difference by which a predicted value is different from the average prediction (mean of all predictions) [123]. It provides contributions that use the same measuring unit as the decision attribute (e.g., if the decision attribute values are prices in euro, the contributions will be expressed in euro as well) which sum up to the aforementioned difference. This way of expression gives a very intuitive representation of the contributions.

Counterfactual explanations aim to determine what is the smallest

possible difference that should be made to the attributes of an instance for which we want to explain the prediction, in order to change the prediction (e.g., to change the decision class). We want to construct artificial instances with a different prediction that are least possibly distant from the given instance. Such artificial instances are called counterfactuals. Examples of methods that create counterfactuals are given in [30, 129]. This approach is again an example of instance-based methods for interpretability that will be discussed later.

1.4.4 Fuzzy logic and interpretability

In the context of interpretability, this thesis wants to contribute to the utilization of fuzzy logic in interpretable machine learning. One of the biggest advantages of fuzzy logic is its possibility to be explained using linguistic expressions. Fuzzy logic expressions and formulas can usually be translated into plain words. This advantage is most visible with fuzzy connectives. Namely, while fuzzy connectives can take rather complex expressions, their interpretation stays clear and simple. There has been a lot of discussion if a fuzzy logic can be used for interpretability purposes [5, 21, 105, 150], and the vast majority of approaches is related to fuzzy rule-based systems. In this thesis, we theoretically explore the concept of fuzzy granules which will serve as building blocks in the development of the rule based systems, in a similar manner as the granularity of rough sets served to the development of the LEM2 and MODLEM algorithms. The development of rule-based systems is, however, out of the scope of this thesis. Apart from the theoretical examination of fuzzy granules, in Chapter 8 we develop an instance-based method that we claim is interpretable. We discuss its interpretability and compare its performance and its interpretablity with other instance-based and locally interpretable models.

Because of this, we provide a brief overview on instance-based methods used for interpretability purposes. We already mentioned instance-based methods that are interpretable up to some extent, like kNN, kFRNN and LVQ, and methods for interpreting black-box models, like prototype-based methods and counterfactuals. Other instance-based methods include adversarial examples and influential instances. Adversarial examples are similar to counterfactuals where the task is to construct an artificial instance, but the role of the instance in this case is adversarial, i.e., it is supposed to deceive the model. We already discussed the example with the STOP sign which is a clear example of an

adversarial instance. Since this method is used to deceive the model as a whole, it is more appropriate to classify it as a global method than as a local method where we include the counterfactuals. However, this method can only give some insights where the model has a flaw, but cannot provide any more general information about the behaviour of the model. The approach with influential instances includes identifying instances in the training dataset for which removal would significantly worsen the prediction performance of the model. By identifying the influential instances, one tries to get to the essence of the model and to better explain its behaviour. The methods for identifying influential instances were discussed in [24, 86].

The reflection of our interpretable model w.r.t. the methods presented here will be discussed in Chapter 8.

1.5 Overview of the dissertation

In this thesis, we integrate ideas and contributions of rough sets, fuzzy sets and machine learning: handling inconsistency and granulation are main contributions of rough sets; the theory of fuzzy sets allows us to use fuzzy relations to model a non-binary interaction among instances; and finally, including statistical/machine learning allows us to make data consistent, incurring the least possible cost (w.r.t. some loss function) using optimization methods.

The contribution of the thesis and its distribution over the chapters is given in Figure 1.1. The major part of the contribution is in providing a strong theoretical background on the existing and new methods used for inconsistency correction together with the desired granular properties. This is shown in yellow rectangles in Figure 1.1. The minor part is developing an instance-based and interpretable classification method which is based on the introduced theory, as well as the discussion of the other possible applications (blue rectangles in Figure 1.1).

Reflecting the described contribution, the thesis is structured as follows:

- Chapter 2: first, we bring to the reader's attention the preliminary material that is needed to understand the concepts developed in this thesis. In particular, we recall the classical indiscernibility-based and dominance-based rough set approaches (IRSA and DRSA); basic notions of fuzzy logic connectives, fuzzy sets and fuzzy relations; statistical learning for monotone classification,

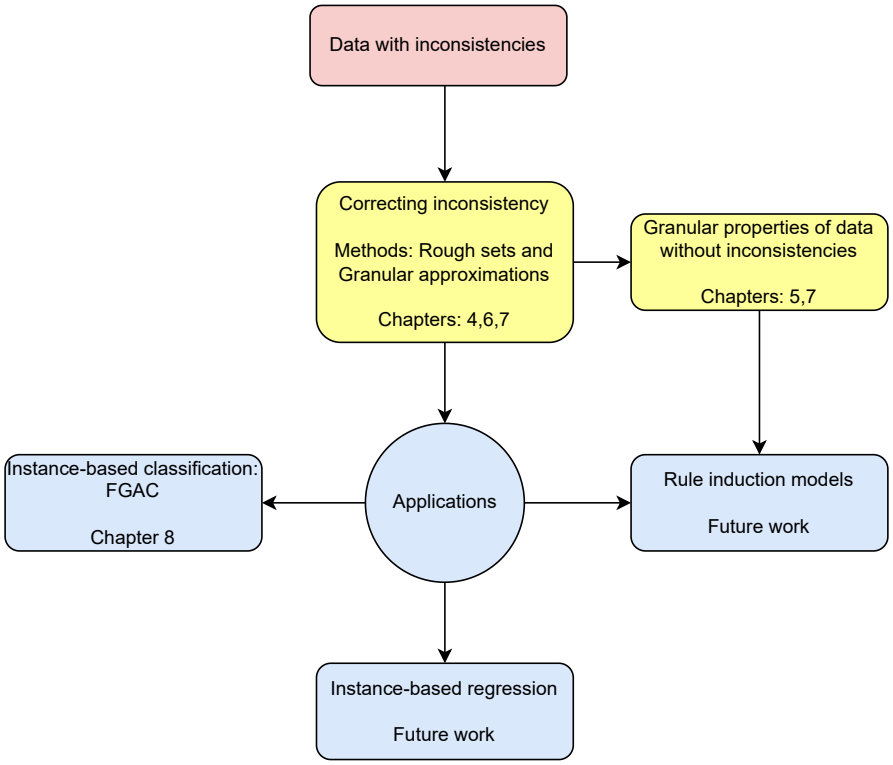

Figure 1.1: Contributions distributed over the chapters of the thesis

and monotone approximation; and finally, the ordered weighted average (OWA) operator that is used in robust extensions of rough set theory.

- In the first part of Chapter 3 we discuss different ways of how specific fuzzy relations (T-preorder and T-equivalence relations) can be constructed and how they are related to metrics and inner products as measures that are widely used to measure how close pairs of instances are. In the second part of the chapter, we formally define the inconsistency problem in data w.r.t. a T-preorder relation and provide classification and regression examples that illustrate all types of inconsistencies that we identified for different crisp or fuzzy relations.

- Chapter 4: for practical purposes, we unify the definitions of IRSA and DRSA into the Preorder-based Rough Set Approach (PRSA)

and propose an integration of PRSA and fuzzy logic. We examine the properties of the integration and propose an extension using Ordered Weighted Average (OWA) operators. We then examine theoretical properties of this hybridisation of OWA operators with fuzzy PRSA, and experimentally compare the robustness of the standard fuzzy DRSA approach with the OWA one.

- Chapter 5: we introduce the concept of a granularly representable set, i.e., a set that can be represented as a union of simple sets or granules. We show the relationship of the new definition with (fuzzy) PRSA, and also explain how the granular approach gives rise to the induction of decision rules. Then, we show that the OWA-based robust extension of the (fuzzy) PRSA model also allows for a granular representation. In particular, we prove that when approximations are defined using a directionally convex t-norm and its residual implicator, the OWA-based lower and upper approximations are definable as unions of fuzzy granules.

- Chapter 6: we introduce a new machine learning method for inconsistency handling with respect to a fuzzy preorder relation. The novel approach is motivated by the existing machine learning approach used for crisp dominance relations described in [89]. We generalize the monotonicity constraints using fuzzy preorder relations, while the ordinal classes are replaced with fuzzy membership degrees, making our approach appropriate for problems where the decision attribute can be modeled using values from the unit interval; concretely, for binary classification and regression problems. The novel approach generalizes the rough set approximations and due to the granular properties of its output, we name the output as *granular approximation*. We provide statistical foundations for our approach, develop optimization procedures that can be used to eliminate inconsistencies, prove some important properties and illustrate the procedures by means of didactic examples.

- Chapter 7: we introduce the concepts of disjoint and adjacent granules and we examine how the new definitions affect the granular approximations. First, we show that the new concepts are important for binary classification problems since they help to keep decision regions separated (disjoint granules) and at the same time to cover as much as possible of the attribute space (adjacent

granules). Then, we formulate an optimization procedure in order to extend granular approximations to the multi-class classification problem leading to the definition of *multi-class granular approximations*. Such an approximation is a union of granules constructed in the way described above; it is a fuzzy set constructed as a conjunction of a fuzzy relation and an association value. These association values can be interpreted as the degree up to which an instance belongs to a certain decision class. Finally, we show how to efficiently calculate multi-class granular approximations for Łukasiewicz fuzzy connectives. We also provide graphical illustrations for a better understanding of the introduced concepts.

- Chapter 8: we design a new classification model called *Fuzzy Granular Approximation Classifier (FGAC)* based on the previously introduced (multi-class) granular approximations. First, we show how FGAC is natively derived from the definition of the granular approximations and develop the classifier for the binary classification case. Then, we extend it for the multi-class classification case. We also propose to utilize OWA operators and develop OWA-based FGAC. The next section deals with ways to improve the time consumption of FGAC at the cost of the precision of the obtained granular approximations. Then, we compare the performance of FGAC with other similar ML models. In the final step, we discuss the interpretability of FGAC and show how the interpretability of the FGAC is more advantageous than that of other ML models.

- Chapter 9: we provide a summary of the thesis, and for each chapter, we separate the previously known results from the original contributions of this thesis. At the end, we discuss possible directions for the continuation of this work.

Chapter 2

Preliminaries

This chapter provides the prior knowledge necessary for understanding the results of this thesis. We discuss the theories of rough sets, fuzzy sets, aggregation operators, statistical learning and the KS approach. We start with rough set theory in Section 2.1 and move on to discuss fuzzy set theory in Section 2.2. In Section 2.3, we outline the basics of statistical learning and then zoom in on statistical learning for monotone classification, developed by Kotłowski and Słowiński. In the last section of this chapter, Section 2.4, we discuss the Ordered Weighted Average (OWA) aggregation operators, used for softening the maximum and minimum.

2.1 Rough set theory

Rough set theory, introduced by Pawlak [103], is a well-known and widely applied mathematical framework for handling inconsistencies in data. In its original formulation, it refers to a set U of instances and an equivalence relation E on U.

More generally, it is possible to replace E by any binary relation on U, not necessarily an equivalence relation. In the next paragraphs, we will review the specific case of an equivalence relation, and of a dominance relation.

2.1.1 Indiscernibility-based rough set Approach

We first recall Pawlak's definition of the Indiscernibility-based Rough Set Approach (IRSA) [103]. Let U be the set of instances and E an equiv-

alence relation on U, which is also called indiscernibility relation. Such relation is

- reflexive: it holds that $(u,u) \in E$,
- symmetric: if $(u,v) \in E$ then $(v,u) \in E$,
- transitive: if $(u,v) \in E$ and $(v,w) \in E$, then also $(u,w) \in E$,

for all $u,v,w \in U$. By $[u]_E$ we denote the equivalence class of E containing element u, i.e.,

$$[u]_E = \{v \in U; (u,v) \in E\}.$$

In the majority of applications, instances are characterised by their values for a number of attributes. Every attribute's domain consists of a finite number of nominal values and every instance $u \in U$ on an attribute $q \in Q$ takes one of those values denoted with $u^{(q)}$. Then, the equivalence relation E is constructed as $(u,v) \in E \Leftrightarrow \forall q \in Q, u^{(q)} = v^{(q)}$. In this case, we say that u and v are indiscernible.

Let $A \subseteq U$ be a subset of instances that belong to the same decision class. The lower and upper approximation of A are defined as:

$$\underline{\text{apr}}_E(A) = \{u \in U : [u]_E \subseteq A\},$$
$$\overline{\text{apr}}_E(A) = \{u \in U : [u]_E \cap A \neq \emptyset\}.$$

2.1.2 Dominance-based rough set approach

In the Dominance-based Rough Set Approach (DRSA), the equivalence relation E is replaced by a dominance relation D which is a preorder, i.e., a reflexive and transitive binary relation on U.

In the applications of DRSA, the set of instances U is described by a finite set of criteria. One criterion's domain consists of a finite number of ordinal values and every instance $u \in U$ on a criterion $q \in Q$ takes one of these values denoted with $u^{(q)}$. These ordinal values generate the preorder relation $D^{(q)}$ on attribute q. Then, the dominance relation D is defined as $(u,v) \in D \Leftrightarrow \forall q \in Q, (u,v) \in D^{(q)}$.

On the other hand, there exists a total order among decision classes, which are denoted by $Cl_t, t \in \{1,\ldots,K\}$. The sets which will be approximated are now upward and downward unions of classes defined respectively as

$$Cl_t^{\geq} = \bigcup_{s \geq t} Cl_s, \quad Cl_t^{\leq} = \bigcup_{s \leq t} Cl_s, \quad t = 1,\ldots,K.$$

$u \in Cl_t^{\geq}$ means that "u belongs at least to Cl_t", while $u \in Cl_t^{\leq}$ means that "u belongs at most to Cl_t". We recall some basic properties of downward and upward unions:

- $Cl_1^{\geq} = Cl_K^{\leq} = U$,
- $Cl_K^{\geq} = Cl_K$ and $Cl_1^{\leq} = Cl_1$,
- for $t = 2, \ldots, K$, it holds: $Cl_{t-1}^{\geq} = U - Cl_t^{\leq}$ and $Cl_{t-1}^{\leq} = U - Cl_t^{\geq}$.

For each $u \in U$ we define the following sets:

- a set $D^+(u)$ of instances dominating u, called dominating set, $D^+(u) = \{v \in U : (v, u) \in D\}$,
- a set $D^-(u)$ of instances dominated by u, called dominated set, $D^-(u) = \{v \in U : (u, v) \in D\}$.

The lower approximation $\underline{\text{apr}}_D(Cl_t^{\geq})$ of $Cl_t^{\geq}$ and the upper approximation $\overline{\text{apr}}_D(Cl_t^{\geq})$ of $Cl_t^{\geq}$ are defined as

$$\underline{\text{apr}}_D(Cl_t^{\geq}) = \{u \in U : D^+(u) \subseteq Cl_t^{\geq}\},$$

$$\overline{\text{apr}}_D(Cl_t^{\geq}) = \{u \in U : D^-(u) \cap Cl_t^{\geq} \neq \emptyset\}.$$

Analogously, we can define the lower and upper approximation of $Cl_t^{\leq}$ as

$$\underline{\text{apr}}_D(Cl_t^{\leq}) = \{u \in U : D^-(u) \subseteq Cl_t^{\leq}\},$$

$$\overline{\text{apr}}_D(Cl_t^{\leq}) = \{u \in U : D^+(u) \cap Cl_t^{\leq} \neq \emptyset\}.$$

2.2 Fuzzy set theory

The definitions and terminology in this section are based on [84].

2.2.1 Fuzzy logic connectives

The fuzzy logic connectives are (unary or binary) operators that are used as fuzzy counterparts of the basic binary logic connectives: conjunction, disjunction, implication and negation.

Triangular norms and copulas

We first recall the notion of a triangular norm, or shortly *t-norm* T, which is an extension of the classical logical conjunction to values in the unit interval. In particular, $T : [0,1]^2 \to [0,1]$ is a binary operator which is commutative, associative, non-decreasing in both arguments, and for which it holds that $\forall x \in [0,1]$, $T(x,1) = x$.

Since a t-norm is associative, we may extend it unambiguously to a $[0,1]^n \to [0,1]$ mapping for any $n > 2$. Some commonly used t-norms are listed in Table 2.1.

Various additional conditions may be imposed on t-norms. We say that $x \in [0,1]$ is a nilpotent element of a t-norm T if there exists a natural number n such that

$$T(\underbrace{x,\ldots,x}_{n \text{ times}}) = 0.$$

A t-norm is called nilpotent if it is continuous and every $x \in (0,1)$ is a nilpotent element. For example, T_L from Table 2.1 is nilpotent while the others are not.

A t-norm is strict if it is continuous and strictly increasing in both arguments. T_P from Table 2.1 is strict while the others are not.

We call a t-norm Archimedean if

$$(\forall (x,y) \in (0,1)^2)(\exists n \geq 2)(T(\underbrace{x,\ldots,x}_{n \text{ times}}) < y).$$

T_P, T_L and T_D from Table 2.1 are Archimedean, while T_M and T_{nM} are not. A continuous Archimedean t-norm T has a unique representation:

$$T(x,y) = f^{-1}(\min(f(x) + f(y), f(0))), \tag{2.1}$$

where f is a decreasing generator, i.e., $f : [0,1] \to \mathbb{R}^+$ is a strictly decreasing continuous $[0,1] \to [0,+\infty]$ mapping for which $f(1) = 0$.

Also, it is known that a t-norm is a continuous Archimedean t-norm if and only if it is either strict or nilpotent.

Name	Definition	R-implicator
Minimum	$T_M(x,y) = \min(x,y)$	$I_{T_M}(x,y) = \begin{cases} 1 & \text{if } x \le y \\ y & \text{otherwise} \end{cases}$
Product	$T_P(x,y) = xy$	$I_{T_P}(x,y) = \begin{cases} 1 & \text{if } x \le y \\ \frac{y}{x} & \text{otherwise} \end{cases}$
Łukasiewicz	$T_L(x,y) = \max(0, x+y-1)$	$I_{T_L}(x,y) = \min(1, 1-x+y)$
Drastic	$T_D(x,y) = \begin{cases} \min(x,y) & \text{if } \max(x,y)=1 \\ 0 & \text{otherwise} \end{cases}$	$I_{T_D}(x,y) = \begin{cases} y & \text{if } x = 1 \\ 1 & \text{otherwise} \end{cases}$
Nilpotent minimum	$T_{nM}(x,y) = \begin{cases} \min(x,y) & \text{if } x+y>1 \\ 0 & \text{otherwise} \end{cases}$	$I_{T_{nM}}(x,y) = \begin{cases} 1 & \text{if } x \le y \\ \max(1-x,y) & \text{otherwise} \end{cases}$

Table 2.1: Some common t-norms and their R-implicators

We call two fuzzy binary operators B^1 and B^2 isomorphic if there exists a bijection $\varphi : [0,1] \to [0,1]$ such that $B^1 = \varphi^{-1}(B^2(\varphi(x), \varphi(y)))$, while unary operators V^1 and V^2 are called isomorphic if $V^1 = \varphi^{-1}(V^2(\varphi(x)))$. Moreover, we write $B^1 \equiv B^2_\varphi$ and $V^1 \equiv V^2_\varphi$.

Proposition 2.2.1. A strict t-norm is isomorphic to T_P while a nilpotent t-norm is isomorphic to T_L.

Given a bijection $\varphi : [0,1] \to [0,1]$, we denote with $T_{L,\varphi}$, defined by

$$T_{L,\varphi}(x,y) = \varphi^{-1}(\max(\varphi(x) + \varphi(y) - 1, 0)) \tag{2.2}$$

the nilpotent t-norm that is isomorphic to T_L with bijection φ and we denote with $T_{P,\varphi}$, defined by

$$T_{P,\varphi}(x,y) = \varphi^{-1}(\varphi(x)\varphi(y)) \tag{2.3}$$

the strict t-norm that is isomorphic to T_P with bijection φ.

Related to the notion of t-norm is that of a *copula*. A (bivariate) copula C is a $[0,1]^2 \to [0,1]$ mapping which satisfies the boundary conditions $\forall x,\ C(0,x) = C(x,0) = 0, C(1,x) = C(x,1) = x$, and the 2-increasingness property: $C(x,y)+C(x',y') \ge C(x',y)+C(x,y')$ for all $x \ge x'$ and $y \ge y'$.

Some t-norms are copulas, while others are not: for example, T_M, T_P and T_L are copulas, while T_D and T_{nM} are not. Vice versa, there also exist copulas which are not t-norms.

Triangular conorms and negators

To model fuzzy logic disjunction, we consider a *t-conorm* $S : [0,1]^2 \to [0,1]$: this is a binary operator which is commutative, associative, non-decreasing in both parameters and for which holds that $\forall x \in [0,1]$, $S(x,0) = x$.

A *negator* (or *fuzzy negation*) $N : [0,1] \to [0,1]$ is a unary non-increasing operator for which it holds that $N(0) = 1$ and $N(1) = 0$. A negator is involutive if $N(N(x)) = x$ for all $x \in [0,1]$. The standard negator is defined as $N_s(x) = 1 - x$.

For a given involutive negator N and a t-norm T, we say that a t-conorm S is the N-dual of T if it holds that $S(x,y) = N(T(N(x),N(y)))$. In this case, the triplet (T,N,S) is called a de-Morgan triplet.

Aggregation operators

Triangular norms and conorms are examples of aggregation operators. A binary aggregation operator $\mathbb{A} : [0,1]^2 \to [0,1]$ (or just aggregation operator) is an operator which is non-decreasing in both arguments, and for which $\mathbb{A}(0,0) = 0$ and $\mathbb{A}(1,1) = 1$. For $x,y \in [0,1]$, an aggregation operator is

- conjunctive if $\mathbb{A}(x,y) \leq \min(x,y)$,
- disjunctive if $\mathbb{A}(x,y) \geq \max(x,y)$,
- averaging if $\min(x,y) \leq \mathbb{A}(x,y) \leq \max(x,y)$.

A t-norm is a conjunctive aggregation operator while a t-conorm is disjunctive.

For a given involutive negator N, we say that aggregation operator $\mathbb{A}$ is N-invariant if

$$\mathbb{A}(x,y) = N(\mathbb{A}(N(x),N(y))). \tag{2.4}$$

It is easy to verify that conjunctive and disjunctive operators cannot be N-invariant.

Implicators

An *implicator* (or *fuzzy implication*) $I : [0,1]^2 \to [0,1]$ is a binary operator which is non-increasing in the first component, non-decreasing in the second one and for which it holds that $I(1,0) = 0$ and $I(0,0) = I(0,1) = I(1,1) = 1$.

Given a t-conorm S and a negator N, the S-implicator induced by S and N is defined as $I(x,y) = S(N(x),y)$.

The residuation property holds for a t-norm T and an implicator I if

$$T(x,y) \leq z \Leftrightarrow x \leq I(y,z). \tag{2.5}$$

It is satisfied if and only if T is left-continuous and I is defined as the residual implicator (R-implicator) of T, that is

$$I_T(x,y) = \sup\{\lambda \in [0,1]; T(x,\lambda) \leq y\}.$$

The very right column of Table 2.1 shows the residual implicators of the corresponding t-norms. Note that all of them, except I_{T_D}, satisfy the residuation property.

Given a $[0,1] \to [0,1]$ bijection φ, the residual implicators of nilpotent and strict t-norms $T_{L,\varphi}$ and $T_{P,\varphi}$ will be denoted by $I_{L,\varphi}$ and $I_{P,\varphi}$.

For implicator I, we define the negator induced by I as $N(x) = I(x,0)$. If I is the R-implicator of t-norm T, we will call the triplet (T,I,N) a residual triplet. If the t-norm from a residual triplet is continuous and Archimedean, then the negator of the triplet is involutive if and only if the t-norm is nilpotent. In such case, the negator has the form

$$N_\varphi(x) = \varphi^{-1}(1 - \varphi(x)).$$

For a residual triplet, the following properties hold for all $x,y,z \in [0,1]$:

- $T(x,y) \leq x \quad \text{and} \quad T(x,y) \leq y,$ (2.6a)
- $I(x,y) \geq y,$ (2.6b)
- $T(x,I(x,y)) \leq y,$ (2.6c)
- $x \leq y \Leftrightarrow I(x,y) = 1,$ (ordering property) (2.6d)
- $T(x,I(y,z)) \leq I(I(x,y),z),$ (2.6e)
- $I(T(x,y),z) = I(x,I(y,z)),$ (2.6f)
- $T(x,N(y)) \leq N(I(x,y))$ (consequence of (2.6e) when $z = 0$), (2.6g)
- $N(T(x,y)) = I(x,N(y))$ (consequence of (2.6f) when $z = 0$). (2.6h)

If residual triplet (T,I,N) is generated by t-norm T, then the residual triplet generated by T_φ is $(T_\varphi, I_\varphi, N_\varphi)$.

A t-norm for which the induced negator of its R-implicator is involutive is called an IMTL t-norm. In Table 2.1, T_L and T_{nM} are IMTL t-norms where the corresponding induced negator is N_s. A residual triplet (T,I,N) that is generated with an IMTL t-norm is called an IMTL triplet. If (T,I,N) is an IMTL triplet, then $(T_\varphi, I_\varphi, N_\varphi)$ is also an IMTL triplet.

For an IMTL triplet, the following properties hold for all $x,y,z \in [0,1]$:

- $I(N(x),N(y)) = I(y,x),$ (2.7a)

- $$T(x, N(y)) = N(I(x, y)). \tag{2.7b}$$

A continuous t-norm is IMTL if and only if it is isomorphic to the Łukasiewicz t-norm. Such t-norm is *strongly max-definable*, i.e., for all $x, y \in [0,1]$ and $I = I_T$, it holds that

$$\max(x, y) = I(I(x, y), y) = I(I(y, x), x). \tag{2.8}$$

The residual triplet generated by $T_{L,\varphi}$, a t-norm isomorphic to T_L, is denoted by $(T_{L,\varphi}, I_{L,\varphi}, N_{L,\varphi})$. Note that $N_L \equiv N_s$.

2.2.2 Fuzzy sets and fuzzy relations

Given a non-empty set U, a fuzzy set A in U is an ordered pair (U, m_A), where $m_A : U \to [0,1]$ is called the membership function and indicates how much an element from U is contained in A. Instead of $m_A(u)$, the membership degree is often written as $A(u)$. If the image of m_A is $\{0,1\}$, we obtain a crisp or classical set. The set of fuzzy sets in U, denoted by $\mathcal{F}(U)$, is thus a superset of $\mathcal{P}(U)$.

For a negator N, the fuzzy complement coA is defined as $coA(u) = N(A(u))$ for $u \in U$. If A is crisp then coA reduces to the standard complement. For $\alpha \in (0,1]$, the α-level set of fuzzy set A is a crisp set defined as $A_\alpha = \{u \in U; A(u) \geq \alpha\}$.

A fuzzy relation $\widetilde{R}$ on U is a fuzzy set on $U \times U$, i.e., a mapping $\widetilde{R} : U \times U \to [0,1]$ which indicates how much two elements from U are related. Some relevant properties of fuzzy relations include:

- $\widetilde{R}$ is reflexive if $\forall u \in U,\ \widetilde{R}(u,u) = 1$.
- $\widetilde{R}$ is symmetric if $\forall u, v \in U,\ \widetilde{R}(u,v) = \widetilde{R}(v,u)$.
- $\widetilde{R}$ is T-transitive w.r.t. t-norm T if $\forall u, v, w \in U$ it holds that $T(\widetilde{R}(u,v), \widetilde{R}(v,w)) \leq \widetilde{R}(u,w)$.

A reflexive and T-transitive fuzzy relation is called a T-preorder, while a symmetric T-preorder is called a T-equivalence. Moreover, if T-equivalence satisfy that $\widetilde{R}(u,v) = 1 \Leftrightarrow u \equiv v$, we call it a T-equality.

If a fuzzy relation $\widetilde{R}$ is T-transitive w.r.t. t-norm T_φ that is isomorphic to T, then the transformed relation $\varphi(\widetilde{R})$ is T-transitive w.r.t. T. The transformed relation is denoted with $\widetilde{R}_\varphi$.

2.3 Kotłowski-Słowiński approach

2.3.1 Statistical learning for monotone classification

A random variable $\mathcal{X}$ is a mapping from a probability space to a certain codomain X. If the codomain is a subset of the real numbers, $\mathcal{X}$ is usually characterized with a cumulative distribution function (CDF) defined as $F_{\mathcal{X}} = P(\mathcal{X} \leq x)$ for $x \in X$. A CDF is a non-decreasing and right-continuous function with codomain $[0, 1]$. If the CDF is continuous then we say that $\mathcal{X}$ is continuous, while if the image of the CDF is a finite set, we say that $\mathcal{X}$ is discrete. Based on the CDF, a quantile function may be defined as follows: $Q_{\mathcal{X}}(p) = \inf\{y; F_{\mathcal{X}}(y) \geq p\}$ for $0 < p < 1$. In other words, if p is in the image of $F_{\mathcal{X}}$, then $Q_{\mathcal{X}}(p)$ is the smallest value for which $P(\mathcal{X} \leq Q_{\mathcal{X}}(p)) = p$. The value $Q_{\mathcal{X}}(\frac{1}{2})$ is called the median of $\mathcal{X}$. The expected value of $\mathcal{X}$ can be expressed using the quantile function [53]:

$$E(\mathcal{X}) = \int_0^1 Q_{\mathcal{X}}(p)dp. \tag{2.9}$$

We say that $\mathcal{X}_1$ *stochastically dominates* $\mathcal{X}_2$ if $F_{\mathcal{X}_1}(x) \geq F_{\mathcal{X}_2}(x)$ for all $x \in X$.

Proposition 2.3.1. [121] For two random variables $\mathcal{X}_1$ and $\mathcal{X}_2$, it holds that

$$\forall x \in X, F_{\mathcal{X}_1}(x) \leq F_{\mathcal{X}_2}(x) \Leftrightarrow \forall p \in (0, 1), Q_{\mathcal{X}_1}(p) \geq Q_{\mathcal{X}_2}(p).$$

The above proposition states that the stochastic dominance can be characterized using quantile functions instead of CDFs.

We now examine the *prediction problem*. Let $\mathcal{X}$ and $\mathcal{Y}$ be two random variables with codomains X and Y respectively. When making predictions, we examine how does $\mathcal{X}$ influence $\mathcal{Y}$. Concretely, we are interested to find a function h such that $h(\mathcal{X})$ is close to $\mathcal{Y}$, i.e., it predicts values of $\mathcal{Y}$ for given values of $\mathcal{X}$. Formally, let $L : Y \times Y \to \mathbb{R}^+$ be a loss function. A prediction problem consists in finding a function $h : X \to Y$ such that the risk

$$R(h) = E(L(\mathcal{Y}, h(\mathcal{X})))$$

is minimized. The optimal h, denoted as h^*, is called the Bayes predictor. The influence of of random variable $\mathcal{X}$ on random variable $\mathcal{Y}$ can be represented by a family of random variables $\mathcal{Y}_{\mathcal{X}=x}$, which stands for variable $\mathcal{Y}$ conditioned by $\mathcal{X} = x$. Such a random variable, for a fixed x, may be described by its CDF:

$$F_{\mathcal{Y}|\mathcal{X}=x}(y) = P(\mathcal{Y} \leq y|\mathcal{X} = x).$$

Searching for an optimal prediction function h in the learning process may be seen as an estimation of certain characteristics of the family of random variables $\mathcal{Y}_{\mathcal{X}=x}$. For example, when the loss function is the squared error loss (also known as quadratic loss or l_2 loss)

$$L_{SEL}(y, \hat{y}) = (y - \hat{y})^2, \tag{2.10}$$

for $y, \hat{y} \in Y$ and $Y = \mathbb{R}$, then the Bayes predictor is $h^*(x) = E(\mathcal{Y}|\mathcal{X} = x)$, i.e., the conditional mean, while if the loss function is the absolute error loss (also known as l_1 loss):

$$L_{AEL}(y, \hat{y}) = |y - \hat{y}|, \tag{2.11}$$

then the Bayes predictor is $h^*(x) = Q_{\mathcal{Y}|\mathcal{X}=x}(\frac{1}{2})$, i.e., the conditional median [16].

In practice, the random variables $\mathcal{X}$ and $\mathcal{Y}$ are unknown as well as their joint distribution and the corresponding Bayes predictor. We only observe their realizations $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$. Our goal is then to minimize the empirical risk:

$$\hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, h(x_i)). \tag{2.12}$$

Minimization of the empirical risk is called *learning* and it basically amounts to an estimation of the unknown Bayes predictor. The empirical risk for the squared error loss is called mean squared error (MSE), while for the absolute error loss is called mean absolute error (MAE). Also, since multiplying an objective function with positive constant does not change the solution, factior $\frac{1}{n}$ is often omitted in 2.12. The examples of Bayes predictors from before show that a Bayes predictor is a characteristic of family $\mathcal{Y}_{\mathcal{X}=x}$ (conditional mean and median in the examples), which means that the learning process leads to an estimation of those characteristics.

Kotłowski and Słowiński [89] introduced a statistical framework for monotone classification. In this case, it is assumed that there is a preorder (dominance) relation $\succeq$ on codomain X of $\mathcal{X}$ while Y consists of a finite number of totally ordered values that distinguish different ordinal classes. Denote these classes by $1, \ldots, K$. The monotonicity constraint states that if $x \succeq x'$ then x has to belong to at least the same class as x'. This is also called the Pareto principle in decision theory. Let $K_{-1} = \{1, \ldots, K-1\}$. In probabilistic terms, the monotonicity constraint

says that $x \succeq x'$ implies

$$\begin{aligned} & \forall k \in K_{-1},\, P(\mathcal{Y} \leq k | \mathcal{X} = x) \leq P(\mathcal{Y} \leq k | \mathcal{X} = x') \\ \Leftrightarrow \quad & \forall k \in K_{-1},\, F_{\mathcal{Y}|\mathcal{X}=x}(k) \leq F_{\mathcal{Y}|\mathcal{X}=x'}(k) \\ \Leftrightarrow \quad & \forall p \in (0,1),\, Q_{\mathcal{Y}|\mathcal{X}=x}(p) \geq Q_{\mathcal{Y}|\mathcal{X}=x'}(p). \end{aligned} \tag{2.13}$$

The previous expression means that the probability that x will be assigned to a class at most k is smaller or equal than that x' will be assigned to the same class. A family $\mathcal{Y}_{\mathcal{X}=x}$ is *monotonically constrained* if (2.13) is satisfied. A prediction function h is called monotone if $x \succeq x' \implies h(x) \geq h(x')$. The goal of monotone classification is to find a proper monotone h under the assumption that the family $\mathcal{Y}_{\mathcal{X}=x}$ is monotonically constrained. Since h, as the output of the learning process, should be as close as possible to the Bayes predictor h^*, we require that h^* is also monotone. Given that the form of h^* depends on the loss function, choosing a proper loss function is crucial for the learning process. A loss function for which the Bayes predictor is monotone is called a monotone loss function. Kotłowski and Słowiński [89] showed that both squared error loss and absolute error loss are monotone loss functions. They also examined a family of monotone loss functions called p-quantile loss (also called pinball loss or linear loss) defined as:

$$L_p(y, \hat{y}) = (y - \hat{y})(p - \mathbf{1}_{y-\hat{y}<0}) = \begin{cases} p|y - \hat{y}| & \text{if } y - \hat{y} > 0, \\ (1-p)|y - \hat{y}| & \text{otherwise.} \end{cases} \tag{2.14}$$

for $p \in [0,1]$, where $\mathbf{1}$ stands for the indicator function. The name p-quantile loss is used since the Bayes predictor for such loss function is the conditional p-quantile $h_p^*(x) = Q_{\mathcal{Y}|\mathcal{X}=x}(p)$. For $p = \frac{1}{2}$ we have that $L_{1/2}$ is equivalent to the absolute error loss. The empirical risk that correspond to the quantile loss is called mean pinball loss (MPL)

For a given loss function L and increasing function φ, we denote scaled loss as:

$$L_\varphi(y, \hat{y}) = L(\varphi(y), \varphi(\hat{y})), \tag{2.15}$$

for all $y, \hat{y} \in Y$. For the p-quantile loss, we have the following important result proved in [89].

Proposition 2.3.2. Let $\varphi : \mathbb{R} \to \mathbb{R}$ be an increasing function. Then the loss functions L_p and their scaled versions $L_{p,\varphi}$ have the same Bayes predictor.

Proposition 2.3.2 states that a different scaling of ordinal classes does not change the Bayes predictor, only the order matters.

Definition 2.3.1. Loss function L is *symmetric* if $L(y,\hat{y}) = L(\hat{y},y)$.

For any increasing function It is easy to verify that $L_{SEL,\varphi}$ and $L_{AEL,\varphi}$ are symmetric loss functions, while $L_{p,\varphi}$ for $p \neq \frac{1}{2}$ is not. However, it can be observed that $L_{p,\varphi}(y,\hat{y}) = L_{1-p,\varphi}(\hat{y},y)$.

Definition 2.3.2. We say that loss function L is *of* $\vee$*-type* if for any real number a, it holds that

- $L(a,a) = 0$,
- functions $L(x,a)$ and $L(a,x)$ are increasing for $x > a$ and
- functions $L(x,a)$ and $L(a,x)$ are decreasing for $x < a$.

The previous definition says that the loss is greater if x is more distant from a. It is easy to verify that the squared error loss and p-quantile loss for $p \in (0,1)$ are of $\vee$-type. The p-quantile loss for $p \in \{0,1\}$ is not of $\vee$-type since $L_0(a,x) = 0$ for $x < a$ and $L_1(a,x) = 0$ for $x > a$.

Definition 2.3.3. A loss function $L : [0,1] \times [0,1] \to \mathbb{R}^+$ is N*-duality preserving* if $L(y,\hat{y}) = L(N(\hat{y}),N(y))$ for N from the residual triplet (T,I,N).

2.3.2 Monotone approximation

In order to incorporate monotonicity constraints into the learning process, the KS approach uses an optimization procedure to "monotonize" data by eliminating inconsistencies. Let $\bar{y}_i, i = 1,\dots,n$, be the observed ordinal labels which do not necessarily satisfy monotonicity constraints due to possible inconsistency, and let $\hat{y}_i, i = 1,\dots,n$, be the values that we want to *learn* and which satisfy the constraints. Then, for a given monotone loss function L, the optimization problem can be formulated as

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} L(\bar{y}_i,\hat{y}_i) \\ \text{subject to} \quad & x_i \succeq x_j \implies \hat{y}_i \geq \hat{y}_j, \quad i,j = 1,\dots,n \\ & \hat{y}_i \in \{0,\dots,K\}, \qquad\qquad i = 1,\dots,n \end{aligned} \tag{2.16}$$

In other words, one wants to calculate new labels that are as close as possible to the original ones w.r.t. loss function L and which satisfy the monotonicity constraints. The obtained labels are called a *monotone approximation* of the original ones. The same authors showed that when L is monotone, then problem (2.16) can be solved using linear programming.

2.4 Ordered Weighted Average

In order to avoid exclusive influence of extreme values (minima and maxima) in decision making, the Ordered Weighted Average (OWA) operator was introduced in [136]. While keeping high influence of the extreme values, the OWA operator also utilizes the values that are non-extreme. It can be seen as a "softer" version of the min and max operators. We recall the definition from [136]. The OWA aggregation of set V of n real numbers with weight vector $W = (w_1, w_2, ..., w_n)$, where $w_i \in [0,1]$ and $\Sigma_{i=1}^{n} w_i = 1$, is given by

$$\mathrm{OWA}_W(V) = \sum_{i=1}^{n} w_i v_{(i)},$$

where $v_{(i)}$ is the i-th largest element in the set V. Different weight vectors are used depending on whether they are used to replace the min or max operator. Those operators can be expressed through OWA operators with the corresponding weights:

$$W_{\min} = (0, \ldots, 0, 1), \quad W_{\max} = (1, 0, \ldots, 0)$$

We call these weights complementary, i.e., it holds that $(W_{\min})_i = (W_{\max})_{n-i+1}$. We denote the complementarity with $W_{\min} = \overline{W_{\max}}$ and $W_{\max} = \overline{W_{\min}}$. Denote with W_L the weights used to replace min and with W_U weights used to replace max. Some well-known weights used in practice are

- additive: $W_L^{add} = (\frac{2}{n(n+1)}, \frac{4}{n(n+1)}, \ldots, \frac{2(n-1)}{n(n+1)}, \frac{2}{n+1})$, $W_U^{add} = \overline{W_L^{add}}$,
- exponential: $W_L^{exp} = (\frac{1}{2^n-1}, \frac{2}{2^n-1}, \ldots, \frac{2^{n-2}}{2^n-1}, \frac{2^{n-1}}{2^n-1})$, $W_U^{exp} = \overline{W_L^{exp}}$,
- inverse additive: $W_L^{invadd} = (\frac{1}{nD_n}, \frac{1}{(n-1)D_n}, \ldots, \frac{1}{2D_n}, \frac{1}{D_n})$, $W_U^{invadd} = \overline{W_L^{invadd}}$ for $D_n = \sum_{i=1}^{n} \frac{1}{i}$.

OWA operators satisfy the monotonicity property:

Proposition 2.4.1. [136] Let V and V' be two sets of n real numbers such that for some permutation σ we have that $\forall i, V_{\sigma(i)} \geq V'_i$. If W is a vector of weights, we have that $\mathrm{OWA}_W(V) \geq \mathrm{OWA}_W(V')$.

It is worth noting that the condition from Proposition 2.4.1 is equivalent to saying that $\forall i, V_{(i)} \geq V'_{(i)}$.

For a given OWA weight vector, we may want to evaluate how well it approximates min and max. For that purpose, the measures *andness* and *orness* are used, where *andness* evaluates how close an aggregation vector is to min while *orness* does the same for max. They are defined as:

$$orness(W) = \frac{1}{n-1}\sum_{i=1}^{n}(w_i(n-i)), \quad andness(W) = 1 - orness(W).$$

The range of the measures is the interval $[0,1]$ where value 1 means that the weight vector is equal to $W_{\min}$ for *andness* or to $W_{\max}$ for *orness*. Also, it holds that if $W_U = \overline{W_L}$, then $andness(W_L) = orness(W_U)$ and vice versa. From [125], we get the evaluations of the measures on the above examples of OWA weights. We have that $andness(W_L^{add}) = \frac{2}{3}$, $andness(W_L^{exp}) = \frac{2^n-n-1}{(n-1)(2^n-1)}$, $andness(W_L^{invadd}) = \frac{n-D_n}{(n-1)D_n}$.

Chapter 3

Fuzzy relations and inconsistency in data

This chapter deals with two crucial components of the thesis: fuzzy relations and data with inconsistencies. In the first part of the chapter, Section 3.1, we discuss different possibilities to construct T-equivalence and T-preorder fuzzy relations. The section contains both well-known results on the topic as well as our original contributions. In the second part of the chapter, Section 3.2, we provide the formal definition of inconsistencies in data with respect to fuzzy relations (T-preorder and T-equivalence). In the same section, we give examples of the four types of inconsistencies in data that are discussed throughout the thesis; inconsistencies w.r.t. an equivalence or indiscernibility relation, a preorder or dominance relation, a T-equivalence relation and a T-preorder relation.

3.1 Construction of fuzzy relations

3.1.1 Triangular similarity and the corresponding dominance relation

In this subsection, we discuss a known T-equivalence relation and provide an example of how to construct a T-preorder relation (i.e., a fuzzy dominance relation), using the existing work on fuzzy rough set theory. Although various authors have worked on proving properties of fuzzy DRSA, none of them constructed a concrete example of a fuzzy dominance relation. We know that for a crisp dominance relation D we may

induce an equivalence or indiscernibility relation E in the following way:

$$(u,v) \in E \Leftrightarrow (u,v) \in D \wedge (v,u) \in D. \tag{3.1}$$

In IRSA, rough sets are defined using the above indiscernibility (or equivalence) relation. In fuzzy IRSA, the indiscernibility relation is replaced with a T-equivalence relation $\widetilde{R}$ which is usually assumed to be reflexive, symmetric and T-transitive for some t-norm T. In the case when data are represented in the form of a decision table, the most used example of such relation is the so-called triangular similarity. For a particular attribute q, it is defined as:

$$\widetilde{E}_q^{\gamma}(u,v) = \max\left(1 - \gamma \frac{|u^{(q)} - v^{(q)}|}{range(q)}, 0\right), \tag{3.2}$$

where $range(q)$ is the range of attribute q and by $\gamma > 0$ we denote the shrinking parameter. Figure 3.1 illustrates this similarity relation. The

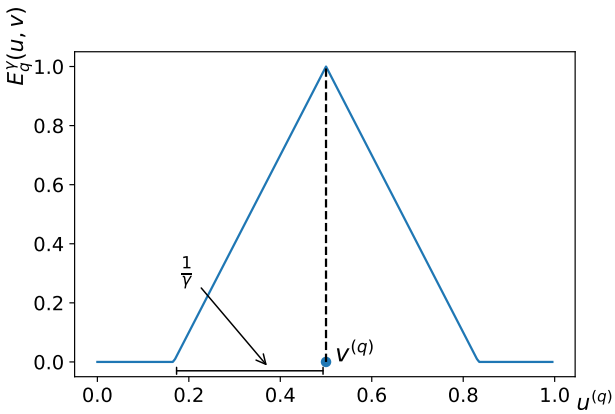

Figure 3.1: Triangular similarity relation on attribute q for pair of instances (u,v).

smaller γ is, the wider the triangle. To construct a T-equivalence relation between instances taking into account the set of attributes Q, we usually use minimum aggregation:

$$\widetilde{E}^{\gamma}(u,v) = \min_{q \in Q} \widetilde{E}_q^{\gamma}(u,v). \tag{3.3}$$

Note that such relations are T-transitive when T is equal to the Łukasiewicz t-norm [32].

To define a T-preorder relation, we want to follow a similar principle as in Eq. (3.1). For a T-preorder relation $\widetilde{D}$ and a similarity relation $\widetilde{E}$, the fuzzy version of (3.1) may be written as:

$$\widetilde{E}(u,v) = T(\widetilde{D}(u,v), \widetilde{D}(v,u)). \tag{3.4}$$

So, we are looking for a T-preorder relation which satisfies (3.4) for the previous definition of $\widetilde{R}$, and which is reflexive and T-transitive, just like the crisp dominance relation. For a particular attribute $q \in Q$, we propose:

$$\widetilde{D}_q^{\gamma}(u,v) = \max\left(\min\left(1 - \gamma \frac{v^{(q)} - u^{(q)}}{range(q)}, 1\right), 0\right). \tag{3.5}$$

An illustration of Eq. (3.5) is given in Figure 3.2. It is easy to check

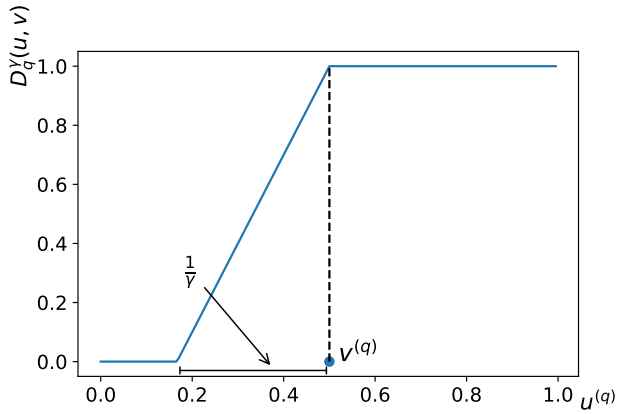

Figure 3.2: An example of the dominance relation on attribute q for pair of instances (u,v).

that (3.4) holds in this case and that $\widetilde{D}_q^{\gamma}$ is reflexive. For T-transitivity we have the following result.

Proposition 3.1.1. Let T be the Łukasiewicz t-norm. Then the fuzzy relation defined by (3.5) is T-transitive, i.e, for elements $u,v,w \in U$ and a attribute q it holds that:

$$\begin{aligned} & T(D_q^{\gamma}(u,v), D_q^{\gamma}(v,w)) \leq D_q^{\gamma}(u,w) \\ \Leftrightarrow \quad & D_q^{\gamma}(u,v) + D_q^{\gamma}(v,w) - 1 \leq D_q^{\gamma}(u,w). \end{aligned}$$

Proof. We denote: $x = 1 - \gamma \frac{v^{(q)} - u^{(q)}}{range(q)}, y = 1 - \gamma \frac{w^{(q)} - v^{(q)}}{range(q)}$. Then, we have to prove that

$$\max(\min(x,1),0) + \max(\min(y,1),0) \leq \max(\min(x+y-1,1),0) + 1.$$

We have three possible cases for x: $x < 0, 0 \leq x \leq 1, x > 1$ and three analogous cases for y, which leads to nine possible cases by their combination. We can reduce this to six cases due to symmetry. Simple verification for each case gives us the desired result. □

To construct a fuzzy dominance relation over all attributes Q, we use min aggregation. We have the following result.

Proposition 3.1.2. Let $D^\gamma(u,v) = \min_{q\in Q} D_q^\gamma(u,v)$. Then D^γ is T-transitive for T being the Łukasiewicz t-norm, i.e. for elements $u, v, w \in U$, it holds that

$$T(D^\gamma(u,v), D^\gamma(v,w)) \leq D^\gamma(u,w).$$

Proof. We have the following:

$$\begin{aligned} T(D^\gamma(u,v), D^\gamma(v,w)) &= T(\min_{q\in Q} D_q^\gamma(u,v), \min_{r\in Q} D_r^\gamma(v,w)) \\ &\leq \min_{q\in Q}\min_{r\in Q} T(D_q^\gamma(u,v), D_r^\gamma(v,w)) \\ &\leq \min_{q\in Q} T(D_q^\gamma(u,v), D_q^\gamma(v,w)) \\ &\leq \min_{q\in Q} D_q^\gamma(u,w) = D^\gamma(u,w). \end{aligned}$$

Here we used monotonicity of T and T-transitivity of D_q^γ. □

3.1.2 T-equivalences based on distances and inner products

Distances

In this section we recall the connection between T-equivalences and distance functions or metrics and discuss which distance-based T-equivalence will be important for us. The relationship between pseudo-metrics and T-equivalences was explored in [32], while the relationship between metrices and T-equalities was explored in [33]. The pseudo-metric d is a mapping $U \times U \to [0, \infty)$ which, for $u, v, w \in U$, satisfies the following:

- $d(u,u) = 0$,
- $d(u,v) = d(v,u)$,
- $d(u,v) + d(v,w) \geq d(u,w)$.

Moreover, a pseudo-metric is a metric if satisfies that $d(u,v) = 0 \Leftrightarrow u \equiv v$. We note that when calculating distances between instances, we deal more with pseudo-metrics since instances that are identically evaluated on all attributes can have distance 0 while still being two separate instances. In table 3.1 we list some well-known metrics (or pseudo-metrics if they are defined on U).

Euclidean distance	$d(u,v) = \sqrt{\sum_{q\in Q}\left(u^{(q)} - v^{(q)}\right)^2}$
Manhattan distance	$d(u,v) = \sum_{q\in Q}\left\lvert u^{(q)} - v^{(q)}\right\rvert$
Chebyshev distance	$d(u,v) = \max_{q\in Q}\left\lvert u^{(q)} - v^{(q)}\right\rvert$

Table 3.1: Well-known metrics

We have the following results.

Proposition 3.1.3. [32] Let T be a continuous Arcimidean t-norm with generator f, and let d be a pseudo-metric on U. Then the binary relation

$$\widetilde{R}(u,v) = f^{-1}(\min(d(u,v), f(0)))$$

is a T-equivalence on U.

Proposition 3.1.4. [33] Let T be a continuous Arcimidean t-norm with generator f, and let d be a metric on U. Then the binary relation

$$\widetilde{R}(u,v) = f^{-1}(\min(d(u,v), f(0)))$$

is a T-equality on U.

We denote T-equivalences and T-equalities obtained using Propositions 3.1.3 and 3.1.4 as distance-based.

We can immediately notice that T_L-equivalence (3.2) together with its aggregation (3.3) is distance-based for Łukasiewicz generator $f(x) = 1-x$ and Chebyshev distance multiplied by γ applied on scaled attributes (by $range(q)$). Moreover, the general parameterized form of a distance-based T_L-equivalence is

$$\widetilde{R}(u,v) = \max\Big(1 - \gamma \cdot d(u,v), 0\Big), \tag{3.6}$$

for pseudo-metric d. Here, using a positive parameter γ is appropriate since if d is a pseudo-metric, then $\gamma \cdot d$ is a pseudo-metric as well. We will

call the T_L-equivalence based on the Euclidean distance as Euclidean similarity and the T_L-equivalence based on the Chebyshev distance as Chebyshev similarity. As stated in Section 3.1.1, Chebyshev similarity is known as triangular similarity from before, and therefore, we may use both terms interchangeably.

At the end, we introduce one interesting metric (or pseudo-metric if it is defined on U) called Mahalanobis distance. It is defined as [94]:

$$d(u,v)_\Sigma = \sqrt{(\mathbf{u}-\mathbf{v})^T \Sigma (\mathbf{u}-\mathbf{v})}, \tag{3.7}$$

where $\mathbf{u}$ is a numerical vector representing the condition attributes of instance u, while Σ is a symmetric and positive-definite matrix. If Σ is an identity matrix, then $d(u,v)_\Sigma$ is equal to the Euclidean distance. This metric will be useful in Chapter 7 where for different matrices Σ we can obtain different shapes of fuzzy granules.

Inner products

Inner products are often used in machine learning to measure similarity among instances. Positive definite kernels (PD kernels), as a generalization of inner products, are used in the development of ML models which construes a family of kernel-based models that include Support Vector Machine, Kernel Logistic Regression and any other method where inner products of instances appear during the optimization procedure [17]. The inner product of two instances u and v, represented with numerical attributes that consisutes vectors $\mathbf{u}$ and $\mathbf{v}$, is defined as

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{q \in Q} u^{(q)} \cdot v^{(q)},$$

while a positive definite kernel is any mapping $k : U \times U \to \mathbb{R}$ for which it holds that

$$\sum_{i=1}^{n} c_i c_j k(u_i, u_j) \geq 0$$

for any $n \in \mathbb{N}$, $u_1, \ldots, u_n \in U$, and $c_1, \ldots, c_n \in \mathbb{R}$. A kernel can be seen as an inner product in a certain Hilbert space that is uniquely determined by the given kernel.

Inner products and kernels with codomain $[0,1]$ were also investigated as T-transitive fuzzy relations [99]. Namely, the authors showed

that for the continuous Archimedean t-norm with generator $f(x) = \arccos(x)$ and with the following form:

$$T_{cos}(x,y) = \max\left(xy - \sqrt{1-x^2}\sqrt{1-y^2}, 0\right),$$

we have that every kernel with a codomain $[0,1]$ is T_{cos}-transitive as a fuzzy relation (bear in mind that every bounded kernel can be extended to have codomain $[0,1]$ by multiplying it with an appropriate constant). However, T_L is not smaller than T_{cos} and we cannot claim that the kernels are also T_L-transitive. However, with an appropriate transformation, we can achieve this. In the same article [99], it was shown that T_{cos} is a nilpotent t-norm and therefore, isomorphic to T_L. The isomorphism in this case is $\varphi_{cos}(x) = 1 - \frac{\arccos(x)}{\pi/2}$. Then, for kernel k with codomain $[0,1]$, we have that transformation $\varphi_{cos}(k)$ is T-transitive w.r.t. T_L.

Since inner products are widely used as a similarity measure, we want to exam under which conditions they can be seen as T_L-equivalences. First, inner products have unbounded domain, but if they are applied on unit vectors, then the codomain is reduced to $[-1,1]$ and the resulting value is the cosine of the angle formed by the two vectors. The second thing is the scaling to $[0,1]$ which is achieved by adding 1 and dividing by two. Then the resulting PD kernel is:

$$k_T(u,v) = 1 + \frac{1 + \frac{\langle \mathbf{u},\mathbf{v}\rangle}{\|\mathbf{u}\|\|\mathbf{v}\|}}{2}.$$

This is indeed a PD kernel since every polynomial transformation of a PD kernel is still a PD kernel [17], and using the transformation with φ_{cos} we obtain T_L-equivalence relation $\varphi_{cos}(k)$.

This relation can be similarly parameterized as above. Namely, in [32], it was shown that if f is a generator of a continuous Archimidean t-norm T, and $\widetilde{R}$ is a T-equivalence, then $f(\widetilde{R})$ is a pseudo-metric. Since $f(x) = 1 - x$ is a generator of T_L, we have that $1 - \varphi_{cos}(k)$ is a pseudo-metric and using similar reasoning as above, we have that $\max(1 - \gamma(1 - \varphi_{cos}(k)), 0)$ is a T_L-equivalence. Therefore, the final form of a parameterized T_L-equivalence based on inner product is:

$$\widetilde{R}^{\gamma}(u,v) = \max\left(1 - \gamma \frac{\arccos\left(1 + \frac{1 + \frac{\langle \mathbf{u},\mathbf{v}\rangle}{\|\mathbf{u}\|\|\mathbf{v}\|}}{2}\right)}{\pi/2}, 0\right). \tag{3.8}$$

This T_L-equivalence will be used to measure similarity between text embeddings in the didactic example from Chapter 8.

3.2 Inconsistencies in data - definition and didactic examples

In this section, we provide the formal definition of data inconsistency and illustrate it on two small examples. Let U be a set of instances, $\tilde{R}$ a T-preorder relation on U and let A be a fuzzy set on U that describes a certain decision using fuzzy membership degrees. We say that a pair $u, v \in U$ is consistent if it holds that

$$T(\widetilde{R}(u,v), A(v)) \leq A(u), \tag{3.9}$$

or equivalently,

$$\widetilde{R}(u,v) \leq I(A(v), A(u)).$$

Both forms are valid due to the residuation property (2.5). In order to better understand Eq. (3.9), we first assume that $\tilde{R}$ and A are crisp, i.e., they take values from $\{0,1\}$. In this case, t-norm T acts as the usual AND logical operator.

If $\tilde{R}$ is symmetric (i.e, it is an equivalence or indiscernibility relation), Eq. (3.9) is interpreted as "If u is indiscernible from v, and v is in A, then u is in A". On the other hand, if $\tilde{R}$ is not symmetric (i.e, it is a preorder or dominance relation), and we assume that class A is more preferred than its complement A^c, Eq. (3.9) is interpreted as "If u is at least as good as v and v is in A, then u is in A".

Now assume that both $\tilde{R}$ and A are fuzzy. If $\tilde{R}$ is a T-equivalence i.e., it is symmetric, we interpret it as a similarity relation which measures how similar two instances are on the $[0,1]$ scale, where 1 stands for indiscernibility while 0 means complete absence of similarity. The interpretation of Eq. (3.9) is "If u is similar to v and v is in A, then u is in A". In this case, "u is in A " is evaluated by means of a membership degree. Analogously, if $\tilde{R}$ is not symmetric, the T-preorder expresses fuzzy dominance. In this case, Eq. (3.9) can be read as "If u is better than or similar to v, and v is in A, then u is in A". Again, the membership to A is expressed in a fuzzy manner.

Next, we show some examples of these four types of inconsistency (symmetric vs. non-symmetric and crisp vs. fuzzy cases) on two datasets; the first involves a binary classification problem, while the second is about regression.

3.2.1 Binary classification

Consider the binary classification problem in Table 3.2, with six instances that represent different customers that applied for a loan. Each of them is described with three attributes: credit card debt, monthly net salary and value of their investment portfolio.

instance	att1 (debt)	att2 (salary)	att3 (portfolio)	decision
1	2200	4200	6000	1
2	7200	2600	7600	1
3	3900	3600	8150	0
4	3900	3600	8150	1
5	10400	3900	9100	0
6	8300	2500	4300	0

Table 3.2: Classification data

The decision attribute expresses if they got the loan (value 1) or not (value 0). We will now identify the four types of inconsistency discussed above in the dataset from Table 3.2.

First, consider the crisp equivalence relation $\tilde{R}$ determined by equality on the condition attributes. Inconsistency w.r.t. $\tilde{R}$ can be observed for instances 3 and 4: they are identically evaluated on all condition attributes, while their decision label is different. In other words, these clients have exactly the same financial parameters, but one client got the loan, while the other one was rejected.

Next, assume $\tilde{R}$ is the following dominance relation determined by the condition attributes: $(u,v) \in \tilde{R}$ as soon as $att1(u) \leq att1(v)$, $att2(u) \geq att2(v)$ and $att3(u) \geq att3(v)$ simultaneously hold (reflecting that $att1$ is a cost-type attribute (the smaller the better) while the others are gain-type attributes (the larger the better)). Instances 2 and 3 are inconsistent w.r.t. this relation: instance 3 is evaluated better than instance 2 on all attributes, but the latter is assigned to a better decision (1) than the former (0). Observe that also instances 3 and 4 are in this relation, which shows that the indiscernibility relation is a particular case of the considered dominance relation.

Now, we move on to involve fuzzy relations. In Table 3.3, we calculate pairwise similarities among instances from Table 3.2 using T_L-equivalence (3.2) for $\gamma = 1$. If we are dealing with a classification prob-

lem, as is the case here, two instances that are assigned to different decision classes are inconsistent as soon as their similarity is bigger than zero, regardless of the choice of the t-norm. For example, if v is instance 2 and u is instance 6, we have that $T_L(\widetilde{R}(u,v), A(v)) = T_L(1, 0.312) = 0.312 > 0 = A(u)$. Therefore, correction of inconsistencies is needed.

	1	2	3	4	5	6
1	1.000	0.059	0.552	0.552	0.000	0.000
2	0.059	1.000	0.412	0.412	0.235	0.312
3	0.552	0.412	1.000	1.000	0.207	0.198
4	0.552	0.412	1.000	1.000	0.207	0.198
5	0.000	0.235	0.207	0.207	1.000	0.000
6	0.000	0.312	0.198	0.198	0.000	1.000

Table 3.3: T_L-equivalence matrix on classification data

In Table 3.4, we calculate the pairwise fuzzy dominance values among instances from Table 3.2 using T_L-preorder (3.5) for $\gamma = 1$. Using the same pair of instances, we can identify the inconsistency w.r.t. the fuzzy dominance relation.

	1	2	3	4	5	6
1	1.000	0.667	0.552	0.552	0.354	1.000
2	0.059	1.000	0.412	0.412	0.235	1.000
3	0.647	1.000	1.000	1.000	0.802	1.000
4	0.647	1.000	1.000	1.000	0.802	1.000
5	0.000	0.610	0.207	0.207	1.000	0.744
6	0.000	0.312	0.198	0.198	0.000	1.000

Table 3.4: T_L-preorder matrix on classification data

To see the added value of using fuzzy relations, note that similarity captures more information on the relationship between instances than indiscernibility. The similarity relation evaluates how close the instances are, while the indiscernibility only determines if the instances have identical condition attributes or not.

When a crisp dominance relation is used, we may face the phenomenon where we have a high number of incomparable instances, i.e., pairs of instances where one instance can be better on one attribute, while the other instance is better on a different attribute. Examples are instances 2 and 5 in Table 3.2, where instance 2 is better on attribute 1

than instance 5 (smaller debt) while instance 5 is better on the two other attributes (higher salary and higher portfolio value). Neither 2 dominates 5 nor 5 dominates 2. A fuzzy dominance relation aids to extract additional information in the form of gradual dominance when we face incomparability. In that way, fuzzy dominance can relax the strictness of the crisp dominance relation.

3.2.2 Regression

In the examples derived from the data from Table 3.2, inconsistencies w.r.t. a fuzzy relation were observed when we deal with a crisp decision. In Table 3.5, we consider a dataset with fuzzy membership values for the decision attribute. This small dataset represents 6 apartments described using 3 condition attributes, while the decision attribute evaluates their expensiveness. The 3 condition attributes are: distance from the nearest public transport station in meters, size of the apartment in square meters and the distance from the nearest grocery store in meters. The decision attribute, expressed with values from interval $[0,1]$, can be obtained using a monotone transformation of the actual prices of the apartments.

instance	att1 distance to transport	att2 size	att3 distance to grocery	decision
1	1200	120	1100	0.770
2	2800	90	900	0.240
3	1900	80	500	0.820
4	2600	60	2200	0.850
5	700	70	3100	0.400
6	3100	50	1400	0.300

Table 3.5: Regression data

Since we are dealing with fuzzy decision labels, it is not possible to consider inconstencies w.r.t. a crisp relation. Therefore, we will identify inconsistencies w.r.t. fuzzy relations. Pairwise evaluations of the T_L-equivalence relation (3.2) on instances from Table 3.5 are given in Table 3.6.

	1	2	3	4	5	6
1	1.000	0.333	0.429	0.143	0.231	0.000
2	0.333	1.000	0.625	0.500	0.125	0.429
3	0.429	0.625	1.000	0.346	0.000	0.500
4	0.143	0.500	0.346	1.000	0.208	0.692
5	0.231	0.125	0.000	0.208	1.000	0.000
6	0.000	0.429	0.500	0.692	0.000	1.000

Table 3.6: T_L-equivalence matrix on regression data

Using the evaluations, if u is instance 2 and v is instance 3, they are inconsistent since $T_L(\widetilde{R}(u,v), A(v)) = T_L(0.625, 0.820) = 0.445 > 0.240 = A(u)$.

Pairwise evaluations of the T_L-preorder relation (3.2) on instances from Table 3.5 are given in Table 3.7. Using the same pair of instances, we have that $T_L(\widetilde{R}(u,v), A(v)) = T_L(0.571, 0.820) = 0.391 > 0.240 = A(u)$.

	1	2	3	4	5	6
1	1.000	0.442	0.442	0.442	0.792	1.000
2	0.429	1.000	0.571	0.976	0.786	1.000
3	0.585	0.817	1.000	0.793	1.000	1.000
4	0.286	0.857	0.429	1.000	0.643	1.000
5	0.390	0.622	0.649	0.598	1.000	1.000
6	0.000	0.000	0.000	0.000	0.351	1.000

Table 3.7: T-preorder matrix on regression data

In the following chapters we show how inconsistencies can be handled using different techniques like rough sets, fuzzy rough sets, OWA-based fuzzy rough sets and granular approximations.

Chapter 4

Preorder-Based Rough Set Approach and its Fuzzy Extensions

We revisit the widely used method for inconsistency handling - rough sets. As already discussed in the Chapter 1, there are two main approaches in the rough set theory that depend on the relation type: the IRSA and the DRSA.

In this chapter, we generalize the definitions of the IRSA and DRSA into the Preorder-based Rough Set Approach (PRSA). Properties that hold for PRSA will automatically transfer to both IRSA and DRSA.

Following the successful hybridisation of fuzzy logic and IRSA, we propose a similar hybridisation of fuzzy logic and PRSA, obtaining fuzzy DRSA as a special case. We prove several properties of fuzzy PRSA that hold for specific fuzzy connectives and we provide counterexamples of the same properties for other fuzzy connectives.

Additionally, we examine the combination of the OWA aggregation operator with fuzzy DRSA. OWA operators were shown to improve IRSA in handling outliers and noisy data [28, 111, 127, 125, 128] by making approximations (and thus also machine learning algorithms that use them) more robust to small changes in the data. Although this goes at the expense of some desirable properties, it was shown, for IRSA at least, that the OWA extension provides the best trade-off between theoretical properties and experimental performance among robust models [35]. In this chapter, we evaluate whether a similar performance may be achieved with fuzzy DRSA. However, the discussion of the connection

between the OWA-based PRSA and the inconsistency handling is left for Chapter 5.

After the introduction of the PRSA model in Section 4.1, in Section 4.2, we consider various possibilities of PRSA fuzzification. In Section 4.3 we present the integration of OWA operators with fuzzy PRSA, while in Section 4.4 we provide an experimental comparison of the robustness between standard and OWA-based fuzzy DRSA. Section 4.6 is reserved for the conclusion, while some specific counterexamples were moved to Section 4.5.

4.1 Preorder-based rough set approach - definition and basic properties

IRSA and DRSA were formally defined in Section 2.1. In the DRSA definition, if we denote $A = Cl_t^{\geq}$ for some t and if D is a symmetric relation, then $D^+(u) = D^-(u)$, and the approximations are reduced to the IRSA definition. So, we may conclude that the DRSA is a generalization of the IRSA. As mentioned, the DRSA is only applied to upward or downward unions, and this specification is purely motivated by the practical applications of the DRSA. As it does not affect any theoretical property of the DRSA approximations, for further use we will introduce the Preorder-based Rough Set Approach (PRSA) in which the DRSA is applied to a general set instead of to an upward or downward union.

The question might be raised whether the PRSA should use the approximations of $A = Cl_t^{\geq}$ or those of $coA = Cl_{t-1}^{\leq}$ from the DRSA definitions. However, we may see that they are in fact equivalent: the approximations of coA may be obtained from those of A by replacing relation D with its inverse relation D^{-1}. Therefore, let R be a preorder relation and let $R^+(u) = \{v \in U, (v,u) \in R\}$ and $R^-(u) = \{v \in U, (u,v) \in R\}$. The lower and upper PRSA approximations of set $A \subseteq U$ are defined as:

$$\begin{aligned} \underline{\text{apr}}_R(A) &= \{u \in U : \ R^+(u) \subseteq A\}, \\ \overline{\text{apr}}_R(A) &= \{u \in U : \ R^-(u) \cap A \neq \emptyset\}. \end{aligned} \tag{4.1}$$

Due to the nature of the definition of PRSA, it automatically inherits all the properties of DRSA. We list the main properties of the PRSA lower and upper approximations [58]:

- (**inclusion**) $\underline{\text{apr}}_R(A) \subseteq A \subseteq \overline{\text{apr}}_R(A)$.

- (**duality**)

$$\underline{\text{apr}}_R(A) = U - \overline{\text{apr}}_{R^{-1}}(coA), \quad \overline{\text{apr}}_R(A) = U - \underline{\text{apr}}_{R^{-1}}(coA).$$

- (**relation monotonicity**) Assume we have another preorder relation $R^* \subseteq R$. Then we have that

$$\underline{\text{apr}}_R(A) \subseteq \underline{\text{apr}}_{R^*}(A), \quad \overline{\text{apr}}_R(A) \supseteq \overline{\text{apr}}_{R^*}(A).$$

We briefly motivate the listed properties. The inclusion property is important as a form to verify that lower and upper approximations stand for certainly consistent and possibly consistent instances respectively. The duality property helps us to deal with classification tasks i.e., when we want to relate the rough approximations of the opposite decision classes. The relation monotonicity is important in attribute selection tasks where the corresponding preorder relation is smaller or larger, w.r.t inclusion, when attributes are added or removed receptively.

Additionally, we also have the properties of exact approximation, decision monotonicity and consistency, idempotence, and interaction between lower and upper approximation:

Proposition 4.1.1. (**exact approximation**)

$$\underline{\text{apr}}_R(A) = A \Leftrightarrow A = \overline{\text{apr}}_R(A),$$

Proof. We have the following sequence of equivalences:

$$\begin{aligned}
\underline{\text{apr}}_R(A) \supseteq A &\Leftrightarrow (\forall u \in A)(R^+(u) \subseteq A) \\
&\Leftrightarrow (\forall u, v \in U)(u \in A \wedge (v,u) \in R \Rightarrow v \in A) \\
&\Leftrightarrow (\forall u, v \in U)(v \notin A \wedge (v,u) \in R \Rightarrow u \notin A) \\
&\Leftrightarrow (\forall u, v \in U)(u \notin A \wedge (u,v) \in R \Rightarrow v \notin A) \\
&\Leftrightarrow (\forall u \in coA)(R^-(u) \subseteq coA) \\
&\Leftrightarrow coA \subseteq \underline{\text{apr}}_{R^{-1}}(coA) \Leftrightarrow coA \subseteq U - \overline{\text{apr}}_R(A) \\
&\Leftrightarrow A \supseteq \overline{\text{apr}}_R(A).
\end{aligned}$$

In the fourth equivalence we just changed the notation; v is replaced with u and u with v, while in the seventh equivalence we used the duality property. Using this with the inclusion property, we complete the proof. □

The previous proposition shows that the approximations can coincide with the approximated set only in the same time.

Proposition 4.1.2. (**decision monotonicity**) For $A^* \subseteq A$, we have that

$$\underline{\text{apr}}_R(A^*) \subseteq \underline{\text{apr}}_R(A), \quad \overline{\text{apr}}_R(A^*) \subseteq \overline{\text{apr}}_R(A).$$

Proof. Obvious from the definition. □

The decision monotonicity proposition is important for upward and downward unions, which among themselves exhibit monotonicity w.r.t. inclusion. Because of that, the decision monotonicity tells us that, e.g., the lower approximation of a smaller upward union will be contained in the lower approximation of a larger one.

The next proposition talks about the consistency of the approximations.

Proposition 4.1.3. (**consistency**) Assume that $(u, v) \in R$. Then we have the following implications.

$$v \in \underline{\text{apr}}_R(A) \Rightarrow u \in \underline{\text{apr}}_R(A), \quad v \in \overline{\text{apr}}_R(A) \Rightarrow u \in \overline{\text{apr}}_R(A).$$

Proof. If $(u, v) \in R$ then we have that $R^+(u) \subseteq R^+(v)$ and $R^-(u) \supseteq R^-(v)$. Putting this into the definitions of the approximations we obtain the result. □

The following two properties are equivalent to the consistency property which will be discussed in more details in the next chapter. Their which the proof can be found in [59].

Proposition 4.1.4. (**idempotence**) It holds that

$$\underline{\text{apr}}_R(\underline{\text{apr}}_R(A)) = \underline{\text{apr}}_R(A), \quad \overline{\text{apr}}_R(\overline{\text{apr}}_R(A)) = \overline{\text{apr}}_R(A).$$

Proposition 4.1.5. (**interaction between lower and upper approximation**) It holds that

$$\overline{\text{apr}}_R(\underline{\text{apr}}_R(A)) = \underline{\text{apr}}_R(A), \quad \underline{\text{apr}}_R(\overline{\text{apr}}_R(A)) = \overline{\text{apr}}_R(A).$$

Example 4.1.1. We provide an example of the application of PRSA and how it handles inconsistencies identified in the data that was introduced in Section 3.2. Since the PRSA deals only with crisp decision values, we only use classification data provided in 3.2. The lower approximation

w.r.t. the indiscernibility relation of the decision class 1 contains instances 1 and 2, while the upper approximation contains instances 1, 2, 3 and 4. We observe that two instances that were inconsistent were either both left from the approximation (the lower case) or were both included in the approximation (upper case), leaving the situation that all pairs of indiscernible instances will have the same decision labels.

If we calculate the approximations w.r.t. a dominance relations, we have that the lower approximation of the more preferred decision 1 contains only instance 1, while the upper approximation of decision 1 contains instances 1, 2, 3 and 4. We again observe that two pairs of inconsistent instances we had $2,3$ and $3,4$ either both belong to the approximation (the upper case) or both do not belong to the approximation (the lower case). Therefore, the pairs of inconsistent instances will have the same decision label.

Here, we can observe why the PRSA can be seen as an extreme way to handle inconsistencies. In the lower and upper approximations, we assign the same decision label to the all pairs of inconsistent instances without leaving possibilities that some pairs obtain one decision label, while the others obtain the opposite one.

4.2 Fuzzy extension of the PRSA

Fuzzy rough sets were introduced for T-equivalence relations and their properties were examined in various publications, see e.g. [35, 41]. In this section, we want to extend those definitions to the fuzzy PRSA and to the fuzzy DRSA as its special case. We want to relax the statement that "u is not worse than v" adding some sort of grading. So, we would like to measure how much the previous statement is true on a scale from 0 to 1. We can interpret this as the credibility of the statement. We start by recalling the approach from Greco et al. [54, 55]. Throughout this section we assume that we are given t-norm T, negator N, t-conorm S, implicator I and T-preorder fuzzy relation $\widetilde{R}$.

If we rewrite the lower approximation definition from Eq. (4.1) as a statement, we have that: "instance u belongs to the lower approximation if $\forall v \in U$, for which $(v,u) \in R$, it holds that $v \in A$". Analogously, "instance u belongs to the upper approximation if $\exists v \in U$, for which $(u,v) \in R$, and it holds that $v \in A$."

In the fuzzy setting, we have to define the membership degree of an instance to the lower and upper approximation. For that purpose, we need to fuzzify the logical quantifiers $\forall$ and $\exists$. We denote these

fuzzy quantifiers as $\text{qua}_\forall$ and $\text{qua}_\exists$. To fuzzify them, two proposals were made. The first one is due to Greco et al. [55] where fuzzy logic connectives are used, i.e., $\text{qua}_\forall = T, \quad \text{qua}_\exists = S$. This option is suitable when the set of instances U is finite as it is in the case of machine learning applications. The second option is proposed by Greco et al. [54] where $\text{qua}_\forall = \inf, \quad \text{qua}_\exists = \sup$. This option is suitable for both cases, when U is finite or infinite and this definition goes in line with the original fuzzy rough approximations proposed by Dubois and Prade [41]. However, in the specific cases where we know that U is finite, we write $(\min, \max)$ instead of $(\inf, \sup)$.

For $(\text{qua}_\forall, \text{qua}_\exists) \in \{(T, S), (\inf, \sup)\}$ we have the following definitions for fuzzy lower and upper approximations:

$$\underline{\text{apr}}_{\widetilde{R}}^{\text{qua}_\forall, I}(A)(u) = \text{qua}_\forall(I(\widetilde{R}(v,u), A(v)); v \in U), \tag{4.2}$$

$$\overline{\text{apr}}_{\widetilde{R}}^{\text{qua}_\exists, T}(A)(u) = \text{qua}_\exists(T(\widetilde{R}(u,v), A(v)); v \in U). \tag{4.3}$$

In order to adopt the new definitions to fuzzy DRSA, we discuss how to construct the fuzzy upward and downward unions. Assume that the decision classes $Cl_t, t \in \{1, \ldots, k\}$ are fuzzy sets with degrees of membership $Cl_t(u)$ for $u \in U$. The value $Cl_t(u)$ provides the credibility that element u belongs to class Cl_t. Greco et al. [55] proposed the concept of cumulative fuzzy upward and downward unions as:

$$Cl_t^{\geq}(u) = \begin{cases} 1, & \text{if} \quad \exists s > t : Cl_s(u) > 0 \\ Cl_t(u) & \text{otherwise} \end{cases}, \tag{4.4}$$

$$Cl_t^{\leq}(u) = \begin{cases} 1, & \text{if} \quad \exists s < t : Cl_s(u) > 0 \\ Cl_t(u) & \text{otherwise} \end{cases}. \tag{4.5}$$

while Du et al. [39] proposed them as fuzzy unions of the classes, i.e.:

$$Cl_t^{\geq}(u) = \max_{s \geq t} Cl_s(u), \quad Cl_t^{\leq}(u) = \max_{s \leq t} Cl_s(u). \tag{4.6}$$

In both of these cases, it holds that for all $u \in U$, $Cl_t^{\geq}(u) \leq Cl_s^{\geq}(u)$ and $Cl_t^{\leq}(u) \geq Cl_s^{\leq}(u)$ if $t \geq s$ and this is the minimal requirement we would ask for any possible definition of $Cl_t^{\geq}(u)$ and $Cl_t^{\leq}(u)$. The corresponding membership degrees to such defined fuzzy sets represent the credibility of the statement: *"u is not worse (not better) than instances from class Cl_t"*.

4.2.1 Properties of the fuzzy PRSA

In this section, we provide the list of properties analogous to those from Section 4.1. Several properties from that list were presented in [55] for $(\mathrm{qua}_\forall, \mathrm{qua}_\exists) = (T, S)$ and in [54] for $(\mathrm{qua}_\forall, \mathrm{qua}_\exists) = (\inf, \sup)$:

- (**inclusion**) $\forall u \in U$:

$$\underline{\mathrm{apr}}_{\widetilde{R}}^{\mathrm{qua}_\forall, I}(A)(u) \leq A(u), \quad \overline{\mathrm{apr}}_{\widetilde{R}}^{\mathrm{qua}_\exists, T}(A)(u) \geq A(u). \tag{4.7}$$

- (**duality**) Let N be an involutive negator for which it holds that $\forall u \in U, coA(u) = N(A(u))$. If (T, S, N) is a de-Morgan triplet and I the S-implicator induced by S and N, or if (T, I, N) is an IMTL triplet, we have that:

$$\begin{aligned} N(\underline{\mathrm{apr}}_{\widetilde{R}}^{\mathrm{qua}_\forall, I}(A)(u)) &= \overline{\mathrm{apr}}_{\widetilde{R}^{-1}}^{\mathrm{qua}_\exists, T}(coA)(u), \\ N(\overline{\mathrm{apr}}_{\widetilde{R}}^{\mathrm{qua}_\exists, T}(A)(u)) &= \underline{\mathrm{apr}}_{\widetilde{R}^{-1}}^{\mathrm{qua}_\forall, I}(coA)(u). \end{aligned} \tag{4.8}$$

- (**relation monotonicity**) For two fuzzy dominance relations $\widetilde{R}$ and $\widetilde{R}^*$ such that $\widetilde{R}^* \subseteq \widetilde{R}$, i.e., $\forall u, v \in U, \widetilde{R}^*(u, v) \leq \widetilde{R}(u, v)$, we have that

$$\begin{aligned} \underline{\mathrm{apr}}_{\widetilde{R}}^{\mathrm{qua}_\forall, I}(A)(u) &\leq \underline{\mathrm{apr}}_{\widetilde{R}^*}^{\mathrm{qua}_\forall, I}(A)(u), \\ \overline{\mathrm{apr}}_{\widetilde{R}}^{\mathrm{qua}_\exists, T}(A)(u) &\geq \overline{\mathrm{apr}}_{\widetilde{R}^*}^{\mathrm{qua}_\exists, T}(A)(u). \end{aligned} \tag{4.9}$$

For $(\mathrm{qua}_\forall, \mathrm{qua}_\exists) = (\inf, \sup)$ we retain the property of exact approximation, as we show below.

Proposition 4.2.1. (**exact approximation**) Let T be a left-continuous t-norm and let I be its R-implicator. Then we have that

$$(\forall u \in U)(\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf, I}(A)(u) = A(u)) \Leftrightarrow (\forall u \in U)(\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup, T}(A)(u) = A(u)).$$

Proof. We will prove the following:

$$(\forall u \in U)(\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf, I}(A)(u) \geq A(u)) \Leftrightarrow (\forall u \in U)(\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup, T}(A)(u) \leq A(u)).$$

The above equivalence, together with the inclusion property, provides the desired result. We have that

$$(\forall u \in U)\,(\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf, I}(A)(u) \geq A(u))$$

$$
\begin{aligned}
&\Leftrightarrow \quad (\forall u \in U)(\inf_{v\in U}(I(\widetilde{R}(v,u),A(v))) \geq A(u)) \\
&\Leftrightarrow \quad (\forall u \in U)(\forall v \in U)(I(\widetilde{R}(v,u),A(v)) \geq A(u)) \\
&\Leftrightarrow \quad (\forall u \in U)(\forall v \in U)(T(\widetilde{R}(v,u),A(u)) \leq A(v)) \\
&\Leftrightarrow \quad (\forall v \in U)(\sup_{u\in U} T(\widetilde{R}(v,u),A(u)) \leq A(v)) \\
&\Leftrightarrow \quad (\forall u \in U)(\sup_{v\in U} T(\widetilde{R}(u,v),A(v)) \leq A(u)) \\
&\Leftrightarrow \quad (\forall u \in U)(\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)(u) \leq A(u)).
\end{aligned}
$$

The third equivalence holds because of the residuation property. In the fifth one, we just change the notation where u is replaced with v and v with u. □

In Example 4.5.1, we provide a counterexample to illustrate that the same property does not hold if $(\mathrm{qua}_\forall, \mathrm{qua}_\exists) = (T, S)$. In Example 4.5.1, the applied implicator is both an S-implicator and an R-implicator. Because of this, we omit using $(\mathrm{qua}_\forall, \mathrm{qua}_\exists) = (T, S)$ and we continue with $(\mathrm{qua}_\forall, \mathrm{qua}_\exists) = (\inf, \sup)$. We want to investigate under which conditions all properties listed above are satisfied. The properties which require additional assumptions on fuzzy logic connectives are duality and exact approximation. We construct counterexamples that illustrate that the exact approximation property does not hold under the assumptions of the duality property and vice versa. In Example 4.5.2, we see that under the assumptions of the duality property we do not necessarily have the exact approximation property while in Example 4.5.3, we may see that R-implicators cannot be used for the duality property in general. So, we conclude that the R-implicators used in the exact approximation property have to be S-implicators in the duality property to have both properties together. The conclusion below unifies these observations.

Proposition 4.2.2. Let T be an IMTL t-norm, I its R-implicator, N the negator induced by I, and S the N-dual of T. Assume also that $(\mathrm{qua}_\forall, \mathrm{qua}_\exists) = (\inf, \sup)$. Then the four properties listed above hold.

We have the following properties of the fuzzy PRSA

Proposition 4.2.3. (**decision monotonicity**) For two decision classes A^* and A for which $A^* \subseteq A$ and for all $u \in U$, we have that

$$
\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A^*)(u) \leq \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u), \quad \overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A^*)(u) \leq \overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)(u),
$$

Proof. Obvious from the definition. □

Proposition 4.2.4. (**consistency**) Let T be a left-continuous t-norm and let I be its R-implicator. We have that

$$\widetilde{R}(u,v) \leq I(\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(v), \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u)),$$
$$\widetilde{R}(u,v) \leq I(\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)(v), \overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)(u)).$$

Proof. We start with the lower approximation. We fix $w \in U$. We have that:

$$\begin{aligned} T(\widetilde{R}(u,v), I(\widetilde{R}(w,v), A(w))) &\leq I(I(\widetilde{R}(u,v), \widetilde{R}(w,v)), A(w))) \\ &\leq I(\widetilde{R}(w,u), A(w)). \end{aligned}$$

The first inequality holds from property (2.6e) while the second one holds the residuation property applied on the T-transitivity of $\widetilde{R}$, i.e., from the following expression

$$T(\widetilde{R}(w,u), \widetilde{R}(u,v)) \leq \widetilde{R}(w,v) \Leftrightarrow \widetilde{R}(w,u) \leq I(\widetilde{R}(u,v), \widetilde{R}(w,v)).$$

From this we may conclude that:

$$\inf_{w_1 \in U} T(\widetilde{R}(u,v), I(\widetilde{R}(w_1,v), A(w_1))) \leq \inf_{w_2 \in U} I(\widetilde{R}(w_2,u), A(w_2)).$$

Since T is increasing, we have that

$$\inf_{w_1 \in U} T(\widetilde{R}(u,v), I(\widetilde{R}(w_1,v), A(w_1))) \geq T(\widetilde{R}(u,v), \inf_{w_1 \in U} I(\widetilde{R}(w_1,v), A(w_1))).$$

Taking this into the previous expression, we obtain:

$$T(\widetilde{R}(u,v), \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(v)) \leq \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u)).$$

Using the residuation principle we obtain the result. For the upper approximation, we may conclude:

$$\begin{aligned} & T(\widetilde{R}(u,v), \widetilde{R}(v,w)) \leq \widetilde{R}(u,w) \\ \Rightarrow & T(T(\widetilde{R}(u,v), \widetilde{R}(v,w)), A(w)) \leq T(\widetilde{R}(u,w), A(w)) \\ \Rightarrow & T(\widetilde{R}(u,v), T(\widetilde{R}(v,w), A(w))) \leq T(\widetilde{R}(u,w), A(w)) \\ \Rightarrow & \sup_{w_1 \in U} T(\widetilde{R}(u,v), T(\widetilde{R}(v,w_1), A(w_1))) \leq \sup_{w_2 \in U} T(\widetilde{R}(u,w_2), A(w_2)) \\ \Rightarrow & T(\widetilde{R}(u,v), \sup_{w_1 \in U} T(\widetilde{R}(v,w_1), A(w_1))) \leq \sup_{w_2 \in U} T(\widetilde{R}(u,w_2), A(w_2)) \end{aligned}$$

$$\Rightarrow T(\widetilde{R}(u,v), \overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)(v)) \leq \overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)(u)$$

The fourth implication holds because T is left-continuous. Applying the residuation principle to the last expression, we obtain the conclusion. □

We provide two more properties as corollaries of the consistency property.

Proposition 4.2.5. (**idempotence**) Let T be a left-continuous t-norm and let I be its R-implicator. It holds that

$$\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)) = \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A), \quad \overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)) = \overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A).$$

Proof. We will prove the proposition for the left expression. The proof for the right one stands by analogy. By the inclusion property we have that

$$\forall u, \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A))(u) \leq \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u).$$

On the other hand, we may apply the residuation principle to the consistency property. For $u \in U$, we have that:

$$\begin{aligned}
&\forall v \in U, \widetilde{R}(v,u) \leq I(\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u), \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(v)) \\
\Leftrightarrow &\forall v \in U, T(\widetilde{R}(v,u), \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u)) \leq \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(v) \\
\Leftrightarrow &\forall v \in U, \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u) \leq I(\widetilde{R}(v,u), \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(v)) \\
\Leftrightarrow &\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u) \leq \inf_{v \in U} I(\widetilde{R}(v,u), \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(v)) \\
\Leftrightarrow &\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u) \leq \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u)),
\end{aligned}$$

which proves the equality. □

Proposition 4.2.6. (**interaction between lower and upper approximation**) Let T be a left-continuous t-norm and let I be its R-implicator. It holds that

$$\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)) = \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A), \quad \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)) = \overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A).$$

Proof. From the inclusion property we have that

$$\forall u \in U, \overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A))(u) \geq \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u).$$

On the other hand, by applying the residuation principle to the consistency property and for $u \in U$, we have

$$\begin{aligned}
&\forall v \in U,\ \widetilde{R}(u,v) \leq I(\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(v), \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u)) \\
\Leftrightarrow &\forall v \in U,\ T(\widetilde{R}(u,v), \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(v)) \leq \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u)) \\
\Leftrightarrow &\sup_{v \in U} T(\widetilde{R}(u,v), \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(v)) \leq \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u)) \\
\Leftrightarrow &\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A))(u) \leq \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u),
\end{aligned}$$

which proves the equality. □

Example 4.2.1. We provide an example of the application of fuzzy PRSA and how it handles inconsistencies identified in the data that was introduced in Section 3.2. First, we calculate the granular approximation of the classification dataset from Table 3.2 using T_L-equivalence relation (3.2), IMTL triplet (T_L, I_L, N_L) and $((\mathrm{qua}_\forall, \mathrm{qua}_\exists) = (\inf, \sup))$. The relation matrix from Table 3.3 is passed together with the decision attribute to formula (3.3). The obtained lower and upper fuzzy rough approximations are given in Table 4.1.

approx. type vs. instance	1	2	3	4	5	6
lower	0.448	0.588	0.000	0.000	0.000	0.000
upper	1.000	1.000	1.000	1.000	0.235	0.312

Table 4.1: The fuzzy PRSA in the classification case for the T_L-equivalence relation

In Table 4.2, we present the calculated fuzzy PRSA approximations using T_L-preorder relation (3.5) while the remaining parameters are the same as in Table 4.1.

approx. type vs. instance	1	2	3	4	5	6
lower	0.353	0.000	0.000	0.000	0.000	0.000
upper	1.000	1.000	1.000	1.000	0.610	0.312

Table 4.2: The fuzzy PRSA in the classification case for the T_L-preorder relation

We note that the pairs of instances are now indeed consistent. Following the example from Section 3.2, where we identified that instances

$u \equiv 6$ and $v \equiv 2$ were inconsistent, using results from Table 4.1, for the lower approximation we obtain $T(\widetilde{R}(u,v),\hat{A}(v)) = T(0.312, 0.588) = 0 \leq 0 = \hat{A}(u)$, i.e., they are now consistent. For the upper approximation, we have that $T(\widetilde{R}(u,v),\hat{A}(v)) = T(0.312, 1) = 0.312 \leq 0.312 = \hat{A}(u)$, i.e., we have the consistency again. If we use the results from Table 4.2, we have that $T(\widetilde{R}(u,v),\hat{A}(v)) = T(0.312, 0) = 0 \leq 0 = \hat{A}(u)$, i.e., they are consistent. For the upper approximation, we have that $T(\widetilde{R}(u,v),\hat{A}(v)) = T(0.312, 1.) = 0.312 \leq 0.312 = \hat{A}(u)$, i.e., we again have the consistency. The values of the fuzzy relations in these examples are obtained from Tables 3.3 and 3.4.

We perform the same calculations for the regression data from Section 3.2 provided in Table 3.5. In order to compute the lower and upper approximations w.r.t. T_L-equivalence relation (3.2), we pass the relation values from Table 3.6 and the decision attribute from Table 3.5 formulas (4.2). The obtained granular approximations are given in Table 4.3.

approx. type vs. instance	1	2	3	4	5	6
lower	0.770	0.240	0.615	0.608	0.400	0.300
upper	0.770	0.445	0.820	0.850	0.400	0.542

Table 4.3: The fuzzy PRSA in the regression case for the T_L-equivalence relation

In Table 4.4, we calculate the fuzzy PRSA approximations using T_L-preorder relation (3.5) while the other parameters are the same as in Table 4.3.

approx. type vs. instance	1	2	3	4	5	6
lower	0.770	0.240	0.615	0.323	0.400	0.240
upper	0.850	0.767	0.850	0.850	0.504	0.642

Table 4.4: The fuzzy PRSA in the regression case for the T_L-preorder relation

We again continue the example from Section 3.2 where we identified that instances $u \equiv 2$ and $v \equiv 3$ are inconsistent. Using values from Table 4.3, for the lower approximation we have that $T(\widetilde{R}(u,v),\hat{A}(v)) = T(0.625, 0.615) = 0.24 \leq 0.24 = \hat{A}(u)$, i.e., they are now consistent. For the upper approximation we have that $T(\widetilde{R}(u,v),\hat{A}(v)) = T(0.625, 0.82) =$

$0.445 \leq 0.445 = \hat{A}(u)$, i.e., we have the consistency again. If we use the calculated values from Table 4.4 for the lower approximation we have that $T(\widetilde{R}(u,v),\hat{A}(v)) = T(0.571, 0.615) = 0.186 \leq 0.24 = \hat{A}(u)$, while for the upper one we have that $T(\widetilde{R}(u,v),\hat{A}(v)) = T(0.571, 0.85) = 0.421 \leq 0.767 = \hat{A}(u)$. In both cases, we corrected the inconsistency.

4.3 Integration with OWA

In this section, we introduce the application of OWA aggregation operators to the fuzzy PRSA and consequently to the fuzzy DRSA. In many practical approaches, we may have outliers: instances that do not follow the general distribution of the data and take some extreme values, e.g., an instance with good values on all considered criteria, assigned to a worse class than many instances getting worse values on these criteria, or conversely, an instance with bad values on all considered criteria assigned to a better class than many of the instances getting better values on these criteria. Because of such instances, many other instances are excluded from the lower approximations of the unions of decision classes they typically belong to. Thus, the lower approximations may be small or even empty. To avoid this, if there is some outlier, we want to reduce its significance in the calculation of the lower approximation. A lot of work has been done to handle such issues for the classical version of DRSA. Some well-known methods include Variable Precision DRSA [75] and Variable Consistency DRSA [61]. Here, we propose a new approach suitable for the fuzzy PRSA and consequently for the fuzzy DRSA, and which is called the OWA approach. OWA operators already showed promising performance in the IRSA, not only for decreasing an outlier's influence in general [35], but also in cases of imbalanced classification [111] and multi-instance learning [128]. OWA operators are applied instead of the fuzzy quantifiers used for a final aggregation in the calculation of the lower and upper approximations.

We recall that the definition and basic properties of the OWA operator is provided in Section 2.4. Assume now that U is finite and that we are given two weight vectors W_L and W_U of size $|U|$. We propose the OWA-based fuzzy PRSA approximations:

$$\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(u) = \mathrm{OWA}_{W_L}(\{I(\widetilde{R}(v,u),A(v)); v \in U\}),$$
$$\overline{\mathrm{apr}}_{\widetilde{R}}^{W_U,T}(A)(u) = \mathrm{OWA}_{W_U}(\{T(\widetilde{R}(u,v),A(v)); v \in U\}).$$

Here, we have more freedom to relax the definition of lower and upper

approximations in order to decrease the significance of possible outliers. If we observe the examples of weights provided in Section 2.4, we can see that the largest weights are multiplied with the possible outliers, but we are including also the values of the other, non-outlying instances into our calculation which is not the case in standard fuzzy PRSA. There we take either the maximum or minimum of the values. An example of how to calculate the OWA-based approximations may be seen in Example 4.5.4.

The OWA-based fuzzy PRSA, as defined in (4.3), does not necessary remove inconsistencies in data. An example for that is provided in 4.5.6. The results on whether OWA-based fuzzy PRSA correct inconsistencies is discussed in Chapter 5.

We will now check if the same properties hold as before. For every $u \in U$, we first notice the following:

$$\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(u) \geq \underline{\mathrm{apr}}_{\widetilde{R}}^{\min,I}(A)(u) \geq \underline{\mathrm{apr}}_{\widetilde{R}}^{T,I}(A)(u),$$

$$\underline{\mathrm{apr}}_{\widetilde{R}}^{W_U,T}(A)(u) \leq \underline{\mathrm{apr}}_{\widetilde{R}}^{\max,T}(A)(u) \leq \underline{\mathrm{apr}}_{\widetilde{R}}^{S,T}(A)(u).$$

Let us now identify some other properties.

Proposition 4.3.1. (**duality**) Let W_L be a weight vector, T, S, N a de-Morgan triplet with $N = N_s$ and let I be the corresponding S-implicator. Let W_U be complementary to W_L. Then, for $u \in U$ it holds that:

$$N(\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(u)) = \overline{\mathrm{apr}}_{\widetilde{R}^{-1}}^{W_U,T}(coA)(u),$$
$$N(\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(u)) = \overline{\mathrm{apr}}_{\widetilde{R}^{-1}}^{W_U,T}(coA)(u).$$

Proof. We will prove just the first expression while the second one will follow by analogy. We fix $u \in U$. Without loss of generality we assume that

$$I(\widetilde{R}(u_1,u),A(u_1)) \geq \cdots \geq I(\widetilde{R}(u_n,u),A(u_n)).$$

Using the assumptions of the proposition, we find:

$$\begin{aligned}
&S(N(\widetilde{R}(u_1,u)),A(u_1)) \geq \cdots \geq S(N(\widetilde{R}(u_n,u)),A(u_n)) \\
\Leftrightarrow\ &N(T(\widetilde{R}(u_1,u),N(A(u_1)))) \geq \cdots \geq N(T(\widetilde{R}(u_n,u),N(A(u_n)))) \\
\Leftrightarrow\ &1-T(\widetilde{R}(u_1,u),1-A(u_1)) \geq \cdots \geq 1-T(\widetilde{R}(u_n,u),1-A(u_n)) \\
\Leftrightarrow\ &T(\widetilde{R}(u_1,u),coA(u_1)) \leq \cdots \leq T(\widetilde{R}(u_n,u),coA(u_n)).
\end{aligned}$$

Therefore, we have that:

$$\begin{aligned}
N(\underline{\text{apr}}_{\widetilde{R}}^{W_L,I}(A)(u)) &= 1 - \sum_{i=1}^{n} (W_L)_i \cdot I(\widetilde{R}(u_i,u), A(u_i)) \\
&= 1 - \sum_{i=1}^{n} (W_L)_i \cdot S(1 - \widetilde{R}(u_i,u), A(u_i)) \\
&= 1 - \sum_{i=1}^{n} (W_L)_i \cdot (1 - T(\widetilde{R}(u_i,u), 1 - A(u_i))) \\
&= \sum_{i=1}^{n} (W_L)_i \cdot T(\widetilde{R}(u_i,u), coA(u_i)) \\
&= \sum_{i=1}^{n} (W_L)_{n-i+1} \cdot T(\widetilde{R}(u_{n-i+1},u), coA(u_{n-i+1})) \\
&= \sum_{i=1}^{n} (W_U)_i \cdot T(\widetilde{R}(u_{n-i+1},u), coA(u_{n-i+1})) = \overline{\text{apr}}_{\widetilde{R}}^{W_U,T}(coA)(u).
\end{aligned}$$

□

Proposition 4.3.2. (**relation monotonicity**) For two fuzzy dominance relations $\widetilde{R}$ and $\widetilde{R}^*$ for which it holds that $\widetilde{R}^* \subseteq \widetilde{R}$, i.e., $\forall u, v \in U, \widetilde{R}^*(u,v) \leq \widetilde{R}(u,v)$, and for any OWA weight vectors W_L and W_U we have that

$$\begin{aligned}
\underline{\text{apr}}_{\widetilde{R}^*}^{W_L,I}(A)(u) &\geq \underline{\text{apr}}_{\widetilde{R}}^{W_L,I}(A)(u), \\
\overline{\text{apr}}_{\widetilde{R}^*}^{W_U,T}(A)(u) &\leq \overline{\text{apr}}_{\widetilde{R}}^{W_U,T}(A)(u).
\end{aligned}$$

Proof. We use the monotonicity of I and T. For $u, v \in U$, we have that

$$\begin{aligned}
I(\widetilde{R}^*(v,u), A(v)) &\geq I(\widetilde{R}(v,u), A(v)), \\
T(\widetilde{R}^*(u,v), A(v)) &\leq T(\widetilde{R}(u,v), A(v)).
\end{aligned}$$

Using Proposition 2.4.1 and the previous inequalities, we complete the proof. □

Proposition 4.3.3. (**decision monotonicity**) For two decision classes A^* and A for which $A^* \subseteq A$ and for all $u \in U$, we have that

$$\begin{aligned}
\underline{\text{apr}}_{\widetilde{R}}^{W_L,I}(A^*)(u) &\leq \underline{\text{apr}}_{\widetilde{R}}^{W_L,I}(A)(u), \\
\overline{\text{apr}}_{\widetilde{R}}^{W_U,T}(A^*)(u) &\leq \overline{\text{apr}}_{\widetilde{R}}^{W_U,T}(A)(u).
\end{aligned}$$

Proof. Using the monotonicity of T and I, for $u, v \in U$ it holds that:

$$I(\widetilde{R}(v,u), A^*(v)) \leq I(\widetilde{R}(v,u), A(v)),$$
$$T(\widetilde{R}(u,v), A^*(v)) \leq T(\widetilde{R}(u,v), A(v)).$$

Using Proposition 2.4.1 and the previous inequalities, we complete the proof. □

Example 4.5.5 shows that the inclusion property does not hold in general. However, we may provide a modification such that the inclusion property holds. We may define:

$$\underline{\text{apr}}_{\widetilde{R}}(A)(u) = \min(A(u), \underline{\text{apr}}_{\widetilde{R}}^{W_L, I}(A)(u)),$$
$$\overline{\text{apr}}_{\widetilde{R}}(A)(u) = \max(A(u), \overline{\text{apr}}_{\widetilde{R}}^{W_U, T}(A)(u)).$$

It is obvious that in this case we will have the inclusion property, however, it is not clear whether this extension is useful in practical applications. The other properties we listed for the standard fuzzy DRSA approach do not hold in general. A counterexample for the exact approximation property is provided in Example 4.5.6 while for the other properties, counterexamples are given in Example 4.5.7.

4.4 Experimental Evaluation

4.4.1 Experimental setup

The robustness of OWA-based fuzzy IRSA has been tested before [35]. In this section, we will compare the robustness of standard fuzzy DRSA and OWA-based fuzzy DRSA. For this purpose, we collected six datasets described in [19] which are used for ordinal classification with monotonicity constraints. Details about these datasets are given in Table 4.5.

name	# of instances	# of condition criteria	# of decision classes
cpu	209	6	4
era	1000	4	9
esl	488	4	9
fame	1328	10	5
lev	1000	4	5
swd	1000	10	4

Table 4.5: Data description

In this experiment, we will consider only the lower approximations of both upward and downward unions, since due to the duality property, the upper approximation performance will be the same. We define the positive region as the fuzzy union of the lower approximations of the upward and downward unions, i.e.,

$$POS_{\widetilde{D},t}(u) = \max(\underline{\text{apr}}_{\widetilde{D}}(Cl_t^{\geq})(u), \underline{\text{apr}}_{\widetilde{D}}(Cl_{t-1}^{\leq})(u)),$$

where $\underline{\text{apr}}$ may stand for either the standard fuzzy DRSA lower approximation or the OWA-based fuzzy DRSA one. We define the positive region for each value $t = 2, 3, \ldots, k$, and we will compare positive region membership values in OWA-based fuzzy DRSA and in standard fuzzy DRSA. If a fuzzy DRSA model is robust, we expect that the positive region does not change drastically when small changes in the data occur. This should be the case when both condition and decision criteria are affected by some fluctuations in data. Therefore, we will consider the cases when the condition and decision criteria are affected by noise separately.

- For the condition criteria, we add Gaussian noise to the data. For a given standard deviation and for each criterion-value pair, we generate a random number from the normal distribution with zero mean and given standard deviation. We add this number to the criterion-value pair. We do this for different standard deviations, which represent the level of noise in the criteria.

- In the decision criterion, we are dealing with ordinal classes. So for a given class in our data, we have three options: it can be increased by one level, decreased by one level or stay the same. Increasing and decreasing may happen with the same, fixed probability. This probability represents the level of noise in the decision criterion. Again, we consider different values in the experiment.

In the experiments we use the fuzzy dominance relation described in Section 3.1.1, with $\gamma = 1$ and where minimum is used as aggregation operator over all criteria. We use Łukasiewicz t-norm $T_L(x, y) = \max(x + y - 1, 0)$ and its R-implicator $I(x, y) = \min(1 - x + y, 1)$. The class sets are constructed such that $Cl_t(u) = 1$ if u belongs to class t, while $Cl_t(u) = 0$ otherwise. Cumulative upward and downward unions are then constructed according to Eq. (4.4). The experiments were performed in the Python programming language together with the Numpy computational library. The seed for the random number generator in Numpy was set to 0.

4.4.2 Perturbations in the condition criteria

We first perform the experiment where we add noise to the criteria. By $\widetilde{D}^*$ we denote the fuzzy dominance relation defined on data with Gaussian noise. We define the Cumulative Mass Difference (CMD) as the value which tells us how much noise affected the positive region of our data. For standard fuzzy DRSA it is defined as:

$$CMD = \sum_{t=2}^{k} \frac{\sum_{u \in U} |POS_{\widetilde{D},t}(u) - POS_{\widetilde{D}^*,t}(u))|}{|U|},$$

while for OWA-based fuzzy DRSA we have that

$$CMD^{OWA} = \sum_{t=2}^{k} \frac{\sum_{u \in U} |POS^{OWA}_{\widetilde{D},t}(u) - POS^{OWA}_{\widetilde{D}^*,t}(u))|}{|U|}.$$

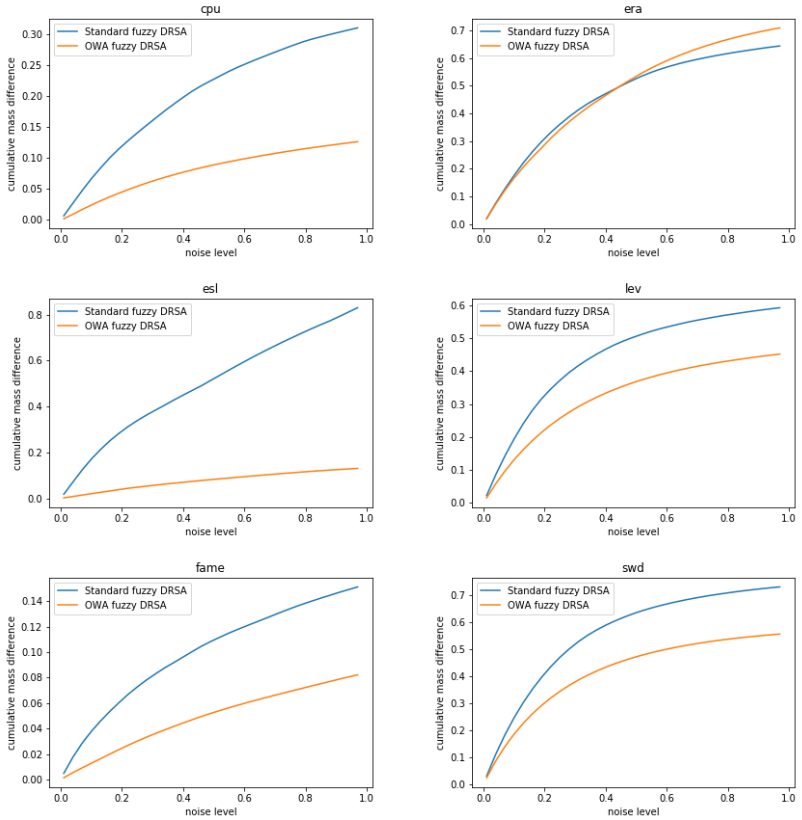

Figure 4.1: CMD with respect to the noise level on condition criteria

Since we want to show the robustness of OWA, we compare CMD and CMD^{OWA} as measures of robustness of standard fuzzy DRSA and OWA-based fuzzy DRSA, where smaller values of CMD mean more robustness. In this particular experiment, additive weights are used for the OWA operator.

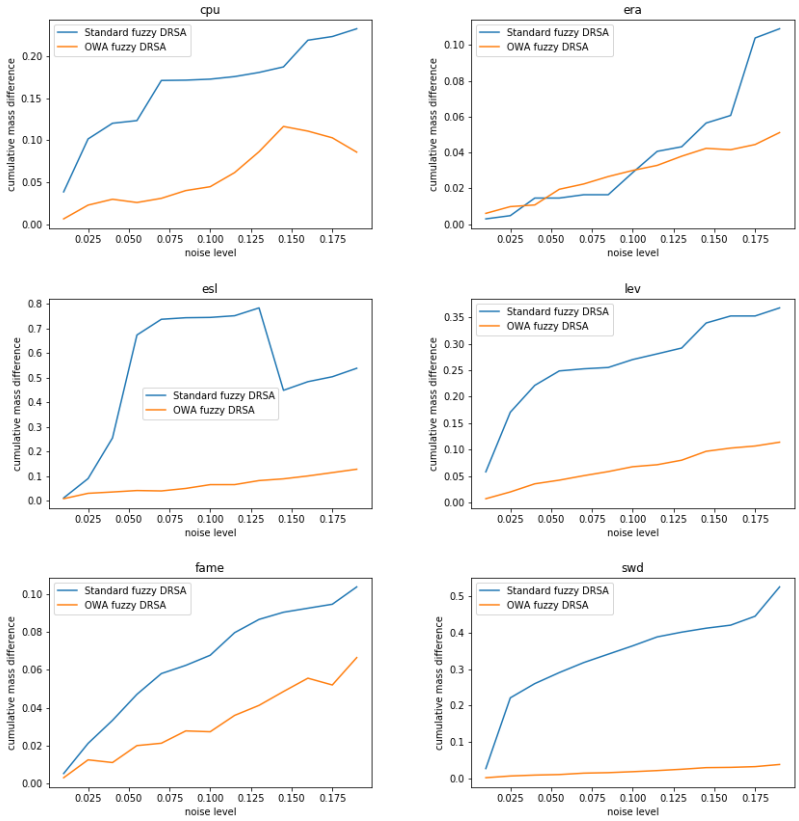

Figure 4.2: CMD w.r.t. the noise level on the decision criterion

In Figure 4.1, we show the dependence of the CMD w.r.t. the noise level, i.e., w.r.t. the standard deviation value given to the Gaussian noise, for the different datasets described above. As we can see, 5 out of 6 images reveal that OWA-based fuzzy DRSA is more robust than standard fuzzy DRSA and that the robustness is increasing as the noise is augmented. Only for the "era" dataset, both approaches show similar robustness with slightly better performance of standard fuzzy DRSA.

4.4.3 Perturbations in the decision criterion

Next, we consider the experiment where noise is added to the decision criterion. By $Cl_{t^*}^{\geq}$ and $Cl_{t^*}^{\leq}$ we denote the respective upward and downward unions after noise is added, while $POS_{D,t^*}(u)$ denotes the positive region with the noisy decision criterion. We define the CMD similarly as above, where $POS_{\widetilde{D}^*,t}(u)$ is replaced with $POS_{D,t^*}(u)$ and $POS_{\widetilde{D}^*,t}^{OWA}(u)$ is replaced with $POS_{D,t^*}^{OWA}(u)$. Again we compare CMD and CMD^{OWA} to show the robustness of the OWA approcach as we did before. Like before, additive weights are used in the OWA operator. In Figure 4.2, we can clearly observe that the OWA-based method outperforms the standard one on 5 out of 6 datasets. As before, the "era" dataset is different, with standard fuzzy DRSA performing better for a small amount of noise, and OWA-based fuzzy DRSA outperforming it for more noisy data.

4.4.4 Using different weights on the "era" dataset

Here, we investigate why the OWA-based fuzzy DRSA fails to outperform the standard one on the “era” dataset. Checking the dataset, we noticed that the positive region calculated with standard fuzzy DRSA has many 0 membership degrees. This indicates a high presence of outliers in the "era" dataset, even without adding any artificial noise. One possibility is that the selection of the weights in this case is not appropriate.

As we stressed above, in the case of additive weights, the largest weights are multiplied with the possible outliers, which still gives some significance to possible outliers and may affect the calculation. To avoid that, we perturb the weight vector W_U by defining a new weight vector W_U' in the following way. Let p be a percentage and let $n_1 = \lfloor pn \rfloor$. We have that:

$$(W_U')_i = \begin{cases} (W_U)_{n-i+1} & \text{if } i = 1,\dots,n_1, \\ (W_U)_{i-n_1} & \text{if } i = n_1+1,\dots,n. \end{cases}$$

In this definition, we take a small percentage ($100p\%$) of values from the right side of the weight vector, flip them and add them to the left side of the weight vector. With this definition, we ensure that the small values from the end of vector W_U are now at the beginning of vector W_U', so the possible outliers will not have such a high significance as they had in W_U. We repeat the experiments for the "era" dataset, now with W_U'

weights instead of W_U and with complementary weight vector $W'_L = \overline{W'_U}$ instead of W_L. We perform these experiments for $p = 0.05$ and $p = 0.1$.

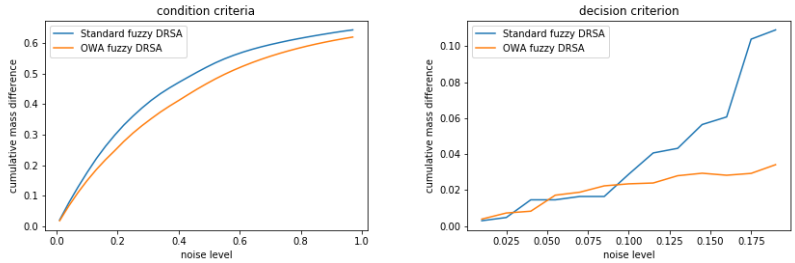

Figure 4.3: CMD w.r.t. the noise level for the "era" dataset and $p = 0.05$.

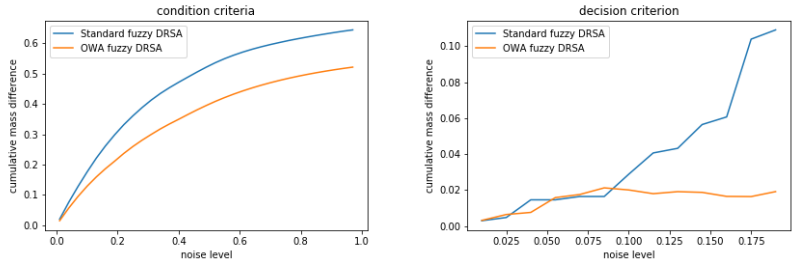

Figure 4.4: CMD w.r.t. the noise level for the "era" dataset and $p = 0.1$.

Figures 4.3 and 4.4 reveal that giving smaller values to the potential outliers improves the performance of OWA-based fuzzy DRSA in both the condition and decision case for the "era" dataset. Moreover, we may see that moving a bigger portion of smaller values to the left side of the weight vector gives even better results, which again indicates the high presence of outliers in the "era" dataset.

4.5 Counterexamples

This section contains a list of counterexamples mentioned earlier in the chapter.

Example 4.5.1. Consider the Łukasiewicz t-norm and its associated R-implicator, i.e., $T(x,y) = \max(x+y-1,0)$ and $I(x,y) = \min(1-x+y,1)$. We induce N from I as $N(x) = 1-x$, which is the standard negator, and we take S to be the N-dual of T, i.e., $S(x,y) = \min(x+y,1)$. Let us assume

that there are two instances a and b such that $A(a) = A(b) = 0.9$. Assume now that the exact approximation property holds, i.e.,

$$\begin{aligned} & (\forall u \in U)(\underline{\text{apr}}_{\widetilde{R}}^{T,I}(A)(u) = A(u)) \\ \Leftrightarrow \quad & (\forall u \in U)\,(T_{v\in U}(I(\widetilde{R}(v,u), A(v))) = A(u)). \end{aligned}$$

We have that

$$\begin{aligned} T_{v\in U}(I(\widetilde{R}(v,a), A(v)) &= T[I(\widetilde{R}(a,a), A(a)), T_{v\neq a}(I(\widetilde{R}(v,a), A(v)))] \\ &= T[A(a), T_{v\neq a}(I(\widetilde{R}(v,a), A(v)))]. \end{aligned}$$

The last expression is equal to $A(a)$ if

$$\begin{aligned} T_{v\neq a}(I(\widetilde{R}(v,a), A(v))) = 1 &\Rightarrow (\forall v \neq a)(I(\widetilde{R}(v,a), A(v)) = 1) \\ &\Rightarrow (\forall v \neq a)(\widetilde{R}(v,a) \leq A(v)). \end{aligned}$$

Now assume $\widetilde{R}(b,a) = 0.9$ which satisfies the condition $\widetilde{R}(b,a) \leq A(b)$. We have that $T(\widetilde{R}(b,a), A(a)) = 0.8$. Then we will have that

$$\begin{aligned} \overline{\text{apr}}_{\widetilde{R}}^{S,T}(A)(b) &= S_{v\in U}(T(\widetilde{R}(b,v), A(v))) \\ &\geq S[T(\widetilde{R}(b,b), A(b)), T(\widetilde{R}(b,a), A(a))] \\ &= S[A(b), T(\widetilde{R}(b,a), A(a))] \\ &= S(0.9, 0.8) = 1 > 0.9 = A(b). \end{aligned}$$

So, we get that for a particular b it holds that $\overline{\text{apr}}_{\widetilde{R}}^{S,T}(A)(b) \neq A(b)$, which is a counterexample to the exact approximation property.

Example 4.5.2. Consider the following de-Morgan triplet: $T(x,y) = \min(x,y), N(x) = 1-x$ and $S(x,y) = \max(x,y)$, with I as the corresponding S-implicator, i.e., $I(x,y) = \max(1-x,y)$. Assume that

$$\begin{aligned} & (\forall u \in U)\,(\underline{\text{apr}}_{\widetilde{R}}^{\min,I}(A)(u) \geq A(u)) \\ \Leftrightarrow & (\forall u \in U)\,(\min_{v\in U}(I(\widetilde{R}(v,u), A(v))) \geq A(u)) \\ \Leftrightarrow & (\forall u \in U)\,(\forall v \in U)\,(I(\widetilde{R}(v,u), A(v)) \geq A(u)). \end{aligned}$$

Let a and b be instances such that $A(a) = 0.4, A(b) = 0.3$ and $\widetilde{R}(b,a) = 0.5$. Then, we have that $I(\widetilde{R}(b,a), A(b)) = I(0.5, 0.3) = 0.5 > 0.4 = A(a)$, so the condition above is satisfied. On the other hand, we have that $T(\widetilde{R}(b,a), A(a)) = T(0.5, 0.4) = 0.4 > 0.3 = A(b)$. Then, we have that

$$\overline{\text{apr}}_{\widetilde{R}}^{\max,T}(A)(b) = \max_{v\in U}(T(\widetilde{R}(b,v), A(v))) \geq T(\widetilde{R}(b,a), A(a)) > A(b),$$

which is a counterexample.

Example 4.5.3. Take de-Morgan triplet $T(x,y) = \min(x,y), S(x,y) = \max(x,y), N = N_s$. Let I be the R-implicator of T, i.e., $I(x,y) = 1$ if $x \leq y$ and $I(x,y) = y$ otherwise. It is obvious that in this case $(T,S) = (\min, \max)$. Assume that for a unique instance b we have that $coA(b) = 0$ and $coA(v) = 1$ for every $v \neq b$. Assume that for some u it holds that $\widetilde{R}(u,b) < 1$. Then, we will have that $\underline{\text{apr}}^{\text{qua}_\forall, I}_{\widetilde{R}^{-1}}(coA)(u) = 0$ since the values of $I(\widetilde{R}(u,v), coA(v))$ are all ones with the one 0 value. On the other hand, we have that

$$\begin{aligned} N(\overline{\text{apr}}^{\text{qua}_\exists, T}_{\widetilde{R}}(A)(u)) &= N(\text{qua}_\exists(T(\widetilde{R}(u,v), A(v)))) \\ &= \text{qua}_\forall(S(N(\widetilde{R}(u,v)), N(A(v)))) \\ &= \text{qua}_\forall(S(N(\widetilde{R}(u,v)), coA(v))) \\ &= S(N(\widetilde{R}(u,b)), coA(b)) = N(\widetilde{R}(u,b)) > 0. \end{aligned}$$

So, we find that for some u, $N(\overline{\text{apr}}^{\text{qua}_\exists, T}_{\widetilde{R}}(A)(u)) > \underline{\text{apr}}^{\text{qua}_\forall, I}_{\widetilde{R}^{-1}}(coA)(u)$, which is a counterexample.

Example 4.5.4. We provide an example of how to calculate OWA-based fuzzy DRSA approximations, and we compare them with the standard fuzzy DRSA approximations. Let us consider the decision table shown in Table 4.6.

obj.	cond1	cond2	decision
a	0.75	0.75	1
b	0.7	0.5	1
c	0.5	0.6	0
d	0.5	0.5	0

Table 4.6: Example of a decision table

We evaluate fuzzy dominance relations among instances of the decision table. We take a fuzzy dominance relation defined as in Section 3.1.1, where $\gamma = 1$ and the aggregation operator is the minimum. We get the matrix shown in Table 4.7.

	a	b	c	d
a	1	1	1	1
b	0.75	1	0.9	1
c	0.75	0.8	1	1
d	0.75	0.8	09	1

Table 4.7: Pairwise evaluations of the fuzzy dominance relation

In the matrix, the value in cell (a,b), for example, represents the value of $\widetilde{R}(a,b)$. For the cumulative upward union, we take the decision vector $(1,1,0,0)$ and its fuzzy set representation, where vector $(1,1,0,0)$ corresponds to fuzzy membership values of instances (a,b,c,d) in fuzzy set A. Then, the complementary set coA is characterized by fuzzy membership values $(0,0,1,1)$. We use additive complementary weight vectors $W_L = (0.1, 0.2, 0.3, 0.4)$ and $W_U = (0.4, 0.3, 0.2, 0.1)$ for the OWA operators. To calculate the lower and upper approximations, we take Łukasiewicz t-norm $T(x,y) = \max(x + y - 1, 0)$ and its R-implicator $I(x,y) = \min(1 - x + y, 1)$. We provide the steps for the calculation of $\underline{\text{apr}}_{\widetilde{R}}^{\min,I}(A)(a)$ and $\underline{\text{apr}}_{\widetilde{R}}^{W_L,I}(A)(a)$. For $v \in \{a,b,c,d\}$ we calculate the values $I(\widetilde{R}(v,a), A(v))$. For $v \in \{a,b,c,d\}$ these values are $\{1,1,0.25,0.25\}$. The standard lower approximation is then calculated by taking the smallest value, i.e., $\underline{\text{apr}}_{\widetilde{R}}^{\min,I}(A)(a) = 0.25$. For the OWA operation, we sort the values in descending order, and apply the weights W_L. We find

$$\underline{\text{apr}}_{\widetilde{R}}^{W_L,I}(A)(a) = 0.1 \cdot 1 + 0.2 \cdot 1 + 0.3 \cdot 0.25 + 0.4 \cdot 0.25.$$

For the other instances, we obtain: $\underline{\text{apr}}_{\widetilde{R}}^{\min,I}(A)(b) = 0.2$, $\underline{\text{apr}}_{\widetilde{R}}^{\min,I}(A)(c) = 0$, $\underline{\text{apr}}_{\widetilde{R}}^{\min,I}(A)(d) = 0$, $\underline{\text{apr}}_{\widetilde{R}}^{W_L,I}(A)(b) = 0.33$, $\underline{\text{apr}}_{\widetilde{R}}^{W_L,I}(A)(c) = 0.2$ and $\underline{\text{apr}}_{\widetilde{R}}^{W_L,I}(A)(d) = 0.167$. For the remaining approximations we get:

$$\begin{aligned}
\overline{\text{apr}}_{\widetilde{R}}^{\max,T}(A) &= \{(a,1),(b,1),(c,0.8),(d,0.8)\},\\
\overline{\text{apr}}_{\widetilde{R}}^{W_U,T}(A) &= \{(a,0.833),(b,0.75),(c,0.65),(d,0.65)\},\\
\underline{\text{apr}}_{\widetilde{R}^{-1}}^{\min,I}(coA) &= \{(a,0),(b,0),(c,0.2),(d,0.2)\},\\
\underline{\text{apr}}_{\widetilde{R}^{-1}}^{W_L,I}(coA) &= \{(a,0.167),(b,0.25),(c,0.35),(d,0.35)\},\\
\overline{\text{apr}}_{\widetilde{R}^{-1}}^{\max,T}(coA) &= \{(a,0.75),(b,0.8),(c,1),(d,1)\},\\
\overline{\text{apr}}_{\widetilde{R}^{-1}}^{W_U,T}(coA) &= \{(a,0.625),(b,0.667),(c,0.8),(d,0.833)\}.
\end{aligned}$$

Example 4.5.5. Assume that A is crisp, which means $A(u) = 1$ if $u \in A$ and $A(u) = 0$ otherwise. Let us compute $I(\widetilde{R}(v,u), A(v))$. If we assume that I is an S-implicator, we have that:

- if $v \in A$, then $I(\widetilde{R}(v,u), A(v)) = 1$,
- if $v \in coA$, then $I(\widetilde{R}(v,u), A(v)) = 1 - \widetilde{R}(v,u)$.

So the values used for OWA aggregation are either $1 - \widetilde{R}(v,u)$ or 1. Assume that $u \notin A \Rightarrow A(u) = 0$. Then, the lower approximation should be 0, but we can always construct a weight vector for the OWA approach to obtain a value different from 0 at the end.

Example 4.5.6. Assume as above that A is crisp and assume that both W_L and W_U do not contain zero weights. Then the evaluations of the implicators will be as above. For the evaluations of the t-norm we have that:

- if $v \in A$, then $T(\widetilde{R}(u,v), A(v)) = \widetilde{R}(u,v)$,
- if $v \in coA$, then $T(\widetilde{R}(u,v), A(v)) = 0$.

Let $u \in A$. Then we have that $\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(u) = A(u) = 1$ if and only if $\forall v \in coA$, $1 - \widetilde{R}(v,u) = 1 \Rightarrow \widetilde{R}(v,u) = 0$. This holds since $\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(u)$ is a convex combination of the elements less or equal than 1 and it can be equal to 1 only if all elements are 1. So, it is possible to satisfy the condition. On the other hand, it is impossible to satisfy $\overline{\mathrm{apr}}_{\widetilde{R}}^{W_U,T}(A)(u) = A(u) = 1$ since we have a convex combination involving zero elements. Thus, we can conclude that the equivalence does not hold in general.

Example 4.5.7. In this counterexample, we use the same data as in Example 4.5.4 as well as the same dominance relation, cumulative upward union and OWA weights. Because of this, the pairwise comparisons will be the same as in Table 4.7. To obtain counterexamples for some properties, we take the left-continuous t-norm $T(x,y) = \sqrt{\max(x^2 + y^2 - 1, 0)}$ and its R-implicator $I(x,y) = \sqrt{\min(1 - x^2 + y^2, 1)}$. We check instances a and d and find that

$$\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(a) = 0.763, \quad \underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(d) = 0.3.$$

We have that $I(\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(a), \underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(d)) = 0.7126$, so

$$\widetilde{R}(d,a) > I(\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(a), \underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(d)),$$

which provides a counterexample for the consistency property. Furthermore, we have that

$$\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A))(a) = 0.7751 \neq \underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(a),$$

which is a counterexample for the idempotence property while it also holds that

$$\overline{\mathrm{apr}}_{\widetilde{R}}^{W_U,T}(\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A))(a) = 0.6374 \neq \underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(a),$$

which is a counterexample for the property about the relation between the lower and upper approximation.

4.6 Conclusion

In this chapter, we presented the main results of the integration of fuzzy set theory and PRSA. We also proposed some improvements using OWA operators to construct a more robust version of fuzzy PRSA. We proved that some properties which hold for classical fuzzy DRSA also hold in the OWA version under specific assumptions. At the end, we empirically showed that OWA-based fuzzy DRSA is indeed more noise tolerant compared to standard fuzzy DRSA.

Chapter 5

Granular Representation of OWA-based Fuzzy Rough Sets

Motivated by the property of (fuzzy) rough sets that they can be represented as a union of simple building blocks i.e., granules, we introduce the concept of a granularly representable (fuzzy) set. While rough sets are constructed based on the assumption of consistency, granularly representable sets are constructed based on the representability of sets by means of granules. However, the two concepts coincide for crisp relations and fuzzy T-preorder relations and this coinciding is being discussed. Moreover, the granular representation naturally gives way to an associated set of decision rules.

We show how granularly representable (fuzzy) sets can be related to (fuzzy) PRSA, i.e., they coincide under specific conditions on the fuzzy connectives. Moreover, as our main contribution, we show that OWA-based fuzzy rough approximations also possess such a granular representation. This holds for a specific type of a fuzzy connectives and for a T-preorder relation, which means that OWA-based PRSA, introduced in Chapter 4, can also be used for inconsistency correction in this case. As a consequence, this robust extension of the fuzzy PRSA can also potentially be used for induction of a set of associated fuzzy rules.

The remainder of this chapter is structured as follows. Section 5.1 introduces the notion of a granularly representable set, and investigates its relationship rough approximations. In this way, we provide a new view on granularity of sets in general, and on the relationship between granularity and rough approximations. In Section 5.2, we define granularly representable fuzzy sets and prove analogous propositions as for

the crisp case. In Section 5.3, we discuss different types of granules and the corresponding rule types that can be induced. Section 5.4 deals with the granularity of OWA-based approximations, while Section 5.5 goes deeper into the topic of characterising convex t-norms which are crucial for the representation. Section 5.6 contains our conclusion.

5.1 Granular view of PRSA

Granular properties of IRSA have been discussed in [141], while a similar analysis was carried out for DRSA in [57]. More recently, the granular representation of DRSA was also studied from the perspective of covering-based rough sets in [34]; in particular, the notion of a definable set was introduced as a union of elementary sets or granules: equivalence classes $[u]_E$ in the case of IRSA, and sets $D^-(u)$ and $D^+(u)$ in the case of DRSA.

In this section, we introduce the notion of a granularly representable set: a set which can be disintegrated into building blocks that are interpreted as human readable rules and we observe the granular properties of the PRSA in view of the new definition. Let U be the set of instances and let $A \subseteq U$. Let R be a preorder relation on U and $R^+(u) = \{v \in U; (v,u) \in R\}$. We say that set A is granularly representable w.r.t. relation R if

$$A = \bigcup_{u \in A} R^+(u).$$

The blocks $R^+(u)$ may be interpreted as indiscernibility rules in the case of IRSA, or monotonic rules in the case of DRSA. Optimality of the rules in the sense of a minimal number of blocks covering A is not guaranteed, and while there exist ways to reduce the number of building blocks of A, this falls outside the scope of this thesis. Here, we focus on the link between granular representability and rough approximations.

Proposition 5.1.1. Set A is granularly representable if and only if $\underline{\text{apr}}_R(A) = A = \overline{\text{apr}}_R(A)$.

Proof. For the right side of the equivalence it is enough to prove or assume that $\underline{\text{apr}}_R(A) = A$ since it holds that $\underline{\text{apr}}_R(A) = A \Leftrightarrow A = \overline{\text{apr}}_R(A)$ due to the exact approximation property.

($\Rightarrow$) Assume that A is granularly representable. For $u \in A$, we have that also $R^+(u) \subseteq A$ which leads to $u \in \underline{\text{apr}}_R(A)$. Hence $A \subseteq \underline{\text{apr}}_R(A)$. Combining this with the inclusion property, we obtain that $\underline{\text{apr}}_R(A) = A$.

($\Leftarrow$) Assume that $\underline{\text{apr}}_R(A) = A$. It holds that

$$u \in A \Rightarrow u \in \underline{\text{apr}}_R(A) \Rightarrow R^+(u) \subseteq A.$$

So we have that $\bigcup_{u \in A} R^+(u) \subseteq A$. On the other hand, from the reflexivity of R it holds that $A \subseteq \bigcup_{u \in A} R^+(u)$. Therefore, A is granularly representable. □

Corollary 5.1.1. $\underline{\text{apr}}_R(A)$ and $\overline{\text{apr}}_R(A)$ are granularly representable sets.

Proof. This follows from the idempotence property of lower and upper approximation. □

Corollary 5.1.2. We may write the rough approximations in the granular form:

$$\underline{\text{apr}}_R(A) = \bigcup \{R^+(u);\, u \in U, R^+(u) \subseteq A\},$$
$$\overline{\text{apr}}_R(A) = \bigcup \{R^+(u);\, u \in A\}.$$

Proof. We have that:

$$\underline{\text{apr}}_R(A) = \bigcup \{R^+(u), u \in \underline{\text{apr}}_R(A)\} = \bigcup \{R^+(u) : u \in U, R^+(u) \subseteq A\},$$

since $u \in \underline{\text{apr}}_R(A) \Leftrightarrow u \in U \wedge R^+(u) \subseteq A$. For the upper approximation, from the granular representability we have that $\overline{\text{apr}}_R(A) = \bigcup \{R^+(u), u \in \overline{\text{apr}}_R(A)\}$. From the inclusion property we know that $A \subseteq \overline{\text{apr}}_R(A)$, so we may conclude that $\bigcup \{R^+(u), u \in A\} \subseteq \bigcup \{R^+(u), u \in \overline{\text{apr}}_R(A)\}$. For the opposite direction we have the following:

$$\begin{aligned}
v \in \bigcup \{R^+(u), u \in \overline{\text{apr}}_R(A)\} &\Leftrightarrow \exists u \in \overline{\text{apr}}_R(A), v \in R^+(u) \\
&\Leftrightarrow \exists u \in U, R^-(u) \cap A \neq \emptyset, v \in R^+(u) \\
&\Leftrightarrow \exists u \in U, \exists w \in A, w \in R^-(u) \wedge v \in R^+(u) \\
&\Leftrightarrow \exists u \in U, \exists w \in A, u \in R^+(w) \wedge v \in R^+(u) \\
&\Rightarrow \exists w \in A, v \in R^+(w) \\
&\Leftrightarrow v \in \bigcup \{R^+(u), u \in A\},
\end{aligned}$$

where for the implication we use the transitivity of R. So, we conclude that also $\bigcup \{R^+(u), u \in \overline{\text{apr}}_R(A)\} \subseteq \bigcup \{R^+(u), u \in A\}$ which gives us the desired result. □

Corollary 5.1.3.

$$R^+(u) \subseteq A \Leftrightarrow R^+(u) \subseteq \underline{\text{apr}}_R(A).$$

Proof. The ($\Leftarrow$) part is obvious because of the inclusion property. ($\Rightarrow$) is a consequence of the definition of the granular representation and Corollary 5.1.2. □

Proposition 5.1.2. Let $A \subseteq U$ and R a preorder on U. The largest granularly representable set contained in A is $\underline{\text{apr}}_R(A)$, while $\overline{\text{apr}}_R(A)$ is the smallest granularly representable set containing A.

Proof. Let B be some granularly representable set containing A. We have that

$$\overline{\text{apr}}_R(A) = \bigcup\{R^+(u) : u \in A\} \subseteq \bigcup\{R^+(u) : u \in B\} = B.$$

Since $\overline{\text{apr}}_R(A)$ is contained in every granularly representable set containing A, $\overline{\text{apr}}_R(A)$ is the smallest such set since it also contains A by the inclusion property. On the other hand, let C be a granularly representable set contained in A. We have that:

$$u \in C \Rightarrow R^+(u) \subseteq C \Rightarrow R^+(u) \subseteq A \Rightarrow u \in \underline{\text{apr}}_R(A).$$

So we conclude that $C \subseteq \underline{\text{apr}}_R(A)$. Since $\underline{\text{apr}}_R(A)$ contains every granularly representable set contained in A, $\underline{\text{apr}}_R(A)$ is the largest such set since it is also contained in A by the inclusion property. □

We have the following result on the relationship between the consistency property and the granular representability.

Proposition 5.1.3. Set $A \subseteq U$ is granularly representable w.r.t. preorder R if and only if it satisfies the consistency property, i.e.,

$$A = \bigcup\{R^+(u); u \in A\} \Leftrightarrow \forall u, v \in U, (v,u) \in R \wedge u \in A \Rightarrow v \in A.$$

Proof. We have the following:

$$\begin{aligned}\bigcup\{R^+(u); u \in A\} \subseteq A \Leftrightarrow & \forall u \in A, R^+(u) \subseteq A \\ \Leftrightarrow & \forall u \in A, \forall v \in U, (v,u) \in R \Rightarrow v \in A \\ \Leftrightarrow & \forall u, v \in U, (v,u) \in R \wedge u \in A \Rightarrow v \in A.\end{aligned}$$

Using the previous equivalence, and the fact that it always holds that $A \subseteq \bigcup\{R^+(u); u \in A\}$ we finish the proof.

□

Example 5.1.1. In this example, we identify the granules that constitute the lower and upper approximations of the data from Section 3.2 provided in Example 4.1.1. If we use the indiscernibility relation, the granules are the equivalence classes of the instances that belong to the approximations. The lower and upper approximations of instances with decision 1 from data in Table 3.2 are $\{1,2\}$ and $\{1,2,3,4\}$ respectively. For those instances, we identify the corresponding equivalence classes. The granules that constitute the lower approximation are $\{1:\{1\},2:\{2\}\}$, while the granules that constitute the upper approximation are $\{1:\{1\},2:\{2\},3:\{3,4\},4:\{3,4\}\}$. In both cases, we listed the instances and the granules that they generate.

If we use the dominance relation, the granules are the dominating set, i.e., all instances that dominate the instance that generate a granule. The lower and upper approximations of instances with decision 1 from data in Table 3.2 are $\{1\}$ and $\{1,2,3,4\}$ respectively. For those instances, we identify the corresponding dominating sets. The single granule that constitutes the lower approximation is $\{1:\{1\}\}$, while the granules that constitute the upper approximation are $\{1:\{1\},2:\{2,3,4\},3:\{3,4\},4:\{3,4\}\}$.

In conclusion, we saw that for every set A and preorder R there is a granular enclosing in the form of rough approximations. They represent families of building blocks which are necessarily (lower approximation) or possibly (upper approximation) contained in A. This may be further translated into possible and certain rules induced from $\overline{\mathrm{apr}}_R(A)$ and $\underline{\mathrm{apr}}_R(A)$, respectively, as done using LEM2 (for IRSA) or DomLEM (for $\overline{\mathrm{DRSA}}$) algorithms.

5.2 Granular representation of fuzzy PRSA

In this section, we extend the granular representation from the previous section to fuzzy sets and relate it to the fuzzy rough set definitions. Let $\widetilde{R}$ be a T-preorder relation for some left-continuous t-norm T, and assume I is its R-implicator. We replace $R^+(u)$ from above with the fuzzy set $\widetilde{R}^+(u)$ where the membership degree of $v \in U$ is given by $\widetilde{R}(v,u)$. Granular properties of fuzzy rough approximations were first introduced in [36], where the authors defined a parameterized family of fuzzy granules:

$$\widetilde{R}^+_\lambda(u) = \{(v, T(\widetilde{R}(v,u),\lambda)); v \in U\}, \tag{5.1}$$

where λ is a real parameter from $[0,1]$. In this definition of a fuzzy granule, we can observe that the construction is motivated by the discussion from Section 1.2.3, i.e., it is a conjunction of the fuzzy set $\{(v, \widetilde{R}(v,u)); v \in U\}$ of instances that relate to the given instance u and the corresponding association value λ. Later on, we will observe that the parameter λ associates a granule to a particular decision.

In the original paper, $\widetilde{R}$ was also symmetric, but later on it was noticed that symmetry does not contribute to the granular properties of fuzzy rough approximations [132]. We observe that Eq. (5.1) is not the only possible way to define fuzzy granules. An alternative was proposed in [44], using implicators and coimplicators. However, in order to extend the granular representation introduced in the previous section, we will focus on the original formula (5.1).

The idea that a set A is granularly representable if it is the union of building blocks $R^+(u)$ with $u \in A$ can be fuzzified by putting $\lambda = A(u)$ in Eq. (5.1).

Definition 5.2.1. We call $A \in \mathcal{F}(U)$ granularly representable if

$$A = \bigcup \{\widetilde{R}^+_{A(u)}(u); u \in U\}. \tag{5.2}$$

We first prove two simple lemmas necessary for further proofs.

Lemma 5.2.1. For $\lambda_1 \leq \lambda_2$ and for $u \in U$ we have that:

$$\widetilde{R}^+_{\lambda_1}(u) \subseteq \widetilde{R}^+_{\lambda_2}(u).$$

Proof. Obvious from the monotonicity of a t-norm. □

Lemma 5.2.2.

$$\widetilde{R}^+_{\lambda}(u) \subseteq A \Leftrightarrow \lambda \leq \underline{\mathrm{apr}}^{\inf,I}_{\widetilde{R}}(A)(u).$$

Proof. We use the residuation property:

$$\begin{aligned}
\widetilde{R}^+_{\lambda}(u) \subseteq A &\Leftrightarrow \forall v \in U,\, T(\widetilde{R}(v,u), \lambda) \leq A(v) \\
&\Leftrightarrow \forall v \in U, \lambda \leq I(\widetilde{R}(v,u), A(v)) \\
&\Leftrightarrow \lambda \leq \inf_{v \in U} I(\widetilde{R}(v,u), A(v)) \Leftrightarrow \lambda \leq \underline{\mathrm{apr}}^{\inf,I}_{\widetilde{R}}(A)(u).
\end{aligned}$$

□

Next, we prove the main result about the granular representability of fuzzy sets.

Proposition 5.2.1. Fuzzy set A is granularly representable w.r.t. relation $\widetilde{R}$ if and only if $\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A) = A = \overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)$.

Proof. As before, for the right side of the equivalence it is enough to prove or assume that $\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A) = A$ since $\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A) = A \Leftrightarrow A = \overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)$ due to the exact approximation property.
($\Rightarrow$) Assume that A is granularly representable. For $v \in U$, we have that

$$A(v) = \sup\{T(\widetilde{R}(v,u), A(u)); u \in U\} = \overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)(v).$$

($\Leftarrow$) Assume that $\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A) = A$. Then, by the same reasoning, we find that A is granularly representable. □

Corollary 5.2.1. Both $\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)$ and $\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)$ are granularly representable fuzzy sets.

Proof. This follows from the idempotence of lower and upper approximations under the considered conditions (Proposition 4.2.5). □

Corollary 5.2.2. We may write the fuzzy rough approximations definitions in the granular form:

$$\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A) = \bigcup\{\widetilde{R}_\lambda^+(u); \widetilde{R}_\lambda^+(u) \subseteq A\}, \quad \overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A) = \bigcup\{\widetilde{R}_{A(u)}^+(u)\}.$$

Proof. For the lower approximation we have that:

$$\begin{aligned}\bigcup\{\widetilde{R}_\lambda^+(u); \widetilde{R}_\lambda^+(u) \subseteq A\} &= \bigcup\{\widetilde{R}_\lambda^+(u); \lambda \leq \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u)\} \\ &= \bigcup\{\widetilde{R}^+_{\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u)}(u)\} = \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A).\end{aligned}$$

The first equality holds because of Lemma 5.2.2 while the second one follows from Proposition 5.2.1. For the upper approximation, it follows directly from the definitions:

$$\begin{aligned}\bigcup\{\widetilde{R}_{A(u)}^+(u)\} &= \{(v, \sup\{T(\widetilde{R}(v,u), A(u)); u \in U\}); v \in U\} \\ &= \overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A).\end{aligned}$$

□

Corollary 5.2.3.

$$\widetilde{R}_\lambda^+(u) \subseteq A \Leftrightarrow \widetilde{R}_\lambda^+(u) \subseteq \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A).$$

Proof. The ($\Leftarrow$) part is obvious since the lower approximation is a subset of the approximated set. ($\Rightarrow$) is a consequence of the definition of the granular representation and Corollary 5.2.2. □

Proposition 5.2.2. Let $A \in \mathcal{F}(U)$ and $\widetilde{R}$ a T-preorder relation. The largest granularly fuzzy representable set contained in A is $\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)$, while $\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)$ is the smallest granularly representable fuzzy set containing A.

Proof. Assume that there is a granularly representable fuzzy set B containing A. We have that:

$$\begin{aligned}\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)(u) &= \sup\{T(\widetilde{R}(u,v), A(v)); v \in U\} \\ &\leq \sup\{T(\widetilde{R}(u,v), B(v)); v \in U\} \\ &= \overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(B)(u) = B(u).\end{aligned}$$

Hence $\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A) \subseteq B$. Since $\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)$ is contained in every granularly representable fuzzy set containing A, $\overline{\mathrm{apr}}_{\widetilde{R}}^{\sup,T}(A)$ is the smallest such fuzzy set since it also contains A by the inclusion property.

On the other hand, assume that C is a granularly representable fuzzy set contained in A. We have that

$$\begin{aligned}\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u) &= \inf\{I(\widetilde{R}(v,u), A(v)), v \in U\} \\ &\geq \inf\{I(\widetilde{R}(v,u), C(v)), v \in U\} \\ &= \underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(C)(u) = C(u).\end{aligned}$$

Since $\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)$ contains every granularly representable fuzzy set contained in A, $\underline{\mathrm{apr}}_{\widetilde{R}}^{\inf,I}(A)(u)$ is the largest such fuzzy set since it is also contained in A by the inclusion property. □

We have the following result about the relationship between the fuzzy consistency property and granular representability.

Proposition 5.2.3. A fuzzy set A in U is granularly representable w.r.t. T-preorder $\widetilde{R}$ if and only if it satisfies the consistency property, i.e.,

$$\begin{aligned}A = \bigcup\{\widetilde{R}^{+}_{A(u)}(u); u \in U\} &\Leftrightarrow \forall u, v \in U,\ \widetilde{R}(v,u) \leq I(A(u), A(v)) \\ &\Leftrightarrow \forall u, v \in U,\ T(\widetilde{R}(v,u), A(u)) \leq A(v).\end{aligned}$$

Proof.

$$\begin{aligned} & A = \bigcup \{\widetilde{R}^{+}_{A(u)}(u); u \in U\} \\ \Leftrightarrow \quad & \forall v \in U, A(v) = \max(T(\widetilde{R}(v,u), A(u)); u \in U) \\ \Leftrightarrow \quad & \forall u, v \in U, A(v) \geq T(\widetilde{R}(v,u), A(u)) \\ \Leftrightarrow \quad & \forall u, v \in U, \widetilde{R}(v,u) \leq I(A(u), A(v)). \end{aligned}$$

The second equivalence holds from the observation that the maximum is reached for $u = v$ due to reflexivity of $\widetilde{R}$. The third equivalence holds from the residuation property. □

Example 5.2.1. In this example, we identify the granules that constitute the upper fuzzy PRSA approximations of the data from Section 3.2 provided in Example 4.2.1. In order to maintain the readability, we will not include the granules of the lower approximations. In tables 5.1, 5.2, 5.3 and 5.4, we have the granules that constitute the upper approximations of the data from Table 3.2 w.r.t. T-equivalence (3.2), the data from Table 3.2 w.r.t. T-preorder (3.5), the data from Table 3.5 w.r.t. T-equivalence (3.2) and the data from Table 3.5 w.r.t. T-preorder (3.5) respectively. The upper approximations are shown in Tables 4.1, 4.2, 4.3, 4.4.

In every row of each table, the first entry represents the generating instance, while the following 6 entries are the fuzzy membership degrees of every instance in the granule generated by the generating instance. For example, In Table 5.3, the first row is a granule generated by instance 1, which is fuzzy set

$$\{(1, 0.77), (2, 0.103), (3, 0.199), (4, 0), (5, 0.001), (6, 0)\}.$$

This granule is obtained in a way that we calculate the T-equivalence values between instance 1 and all other instances where the resulting fuzzy set is:

$$\{(1,1), (2, 0.333), (3, 0.429), (4, 0.143), (5, 0.231), (6, 0)\}, \tag{5.3}$$

and then we evaluate T_L operator to the fuzzy degrees from (5.3) and the upper approximation membership of instance 1 which is 0.77. We can also observe that if we calculate the maximum of every row, we will get the upper approximation degrees, i.e., the union of granules is indeed the upper approximation.

	1	2	3	4	5	6
1	1	0.059	0.552	0.552	0	0
2	0.059	1	0.412	0.412	0.235	0.312
3	0.552	0.412	1	1	0.207	0.198
4	0.552	0.412	1	1	0.207	0.198
5	0	0	0	0	0.235	0
6	0	0	0	0	0	0.312

Table 5.1: Granules of the upper approximation from Table 4.1

	1	2	3	4	5	6
1	1	0.667	0.552	0.552	0.354	1
2	0.059	1	0.412	0.412	0.235	1
3	0.647	1	1	1	0.802	1
4	0.647	1	1	1	0.802	1
5	0	0.22	0	0	0.610	0.354
6	0	0	0	0	0	0.312

Table 5.2: Granules of the upper approximation from Table 4.2

	1	2	3	4	5	6
1	0.77	0.103	0.199	0	0.001	0
2	0	0.445	0.07	0	0	0
3	0.249	0.445	0.82	0.166	0	0.32
4	0	0.35	0.196	0.85	0.058	0.542
5	0	0	0	0	0.4	0
6	0	0	0.042	0.235	0	0.542

Table 5.3: Granules of the upper approximation from Table 4.3

	1	2	3	4	5	6
1	0.850	0.773	0.619	0.85	0.642	0.85
2	0.1	0.767	0.392	0.683	0	0.767
3	0.279	0.707	0.85	0.85	0.350	0.85
4	0	0.35	0.196	0.85	0.058	0.542
5	0	0	0	0.158	0.504	0
6	0	0.07	0.142	0.433	0	0.642

Table 5.4: Granules of the upper approximation from Table 4.4

As we saw before, for any fuzzy set A, there is a fuzzy granular enclosing composed of fuzzy rough approximations. With this, we obtain families of fuzzy building blocks which may be interpreted as certain and possible fuzzy rules. Concrete examples of fuzzy rough rule induction may be found in [79, 149].

5.3 Granules and their interpretation

As we mentioned before, granules are important from the perspective of rule induction. We keep granules simple, such that one granule corresponds to one rule. Since a granularly representable set is a union of granules, it can be seen as a union of rules, so it is fully readable by a human. With granules in the PRSA and fuzzy PRSA we are able to identify four types of rules: two types for the crisp case (IRSA and DRSA) and two types for the fuzzy case (fuzzy IRSA and fuzzy DRSA). The lower approximations generate certain rules, while the upper approximations generate possible rules. For every type of granule, we provide directions how to construct a rule in a classification problem, and an example of such rule. We provide examples that correspond to the granules of the upper approximations from Examples 5.1.1 and 5.2.1 of classification data from Table 3.2. Since the rules are extracted from the upper approximations, we consider them as possible rules.

We first discuss IRSA granules and rules. Assume that instancess are described by attributes as explained in Section 1.1.1.

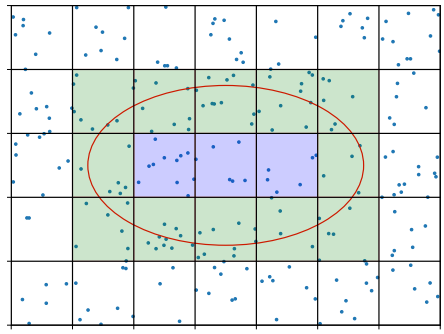

Figure 5.1: Crisp approximations with equivalence relation

In Figure 5.1, we show an example of binary classification (number of decision classes is 2) where 250 instances (points) are separated by the elliptical curve into interior and exterior classes. The equivalence classes in the set of instances are represented by the squares in the figure (35 equivalence classes). The lower approximation of the interior class is marked with light blue color, while its upper approximation is the union of light green and light blue squares. The approximations are granularly representable sets so they are equal to the union of the equivalence classes of their instances. We notice that we can choose one granule per representative element for each equivalence class. Therefore, the interior can be represented as a union of three classes as we can see in the figure. Each such granule can be seen as a rule. Since equivalence classes consist of instances with equal values on all attributes, the rules have the following form:

$$\text{IF att}_1 = \text{val}_1 \text{ AND ... AND att}_m = \text{val}_m \text{ THEN decision is dec.}$$

Here "att", "val" and "dec" are abbreviations for "attribute", "value" and "decision". Here "dec" stands for a decision class which is assigned to the classified instance.

We provide an example of such a rule from the granule that correspond to instance 3 in Example 5.1.1:

$$\text{IF debt} = 3900 \text{ AND salary} = 3600 \text{ AND portfolio} = 8150$$
$$\text{THEN decision is } 1.$$

This is a possible rule which is fulfilled by the instances that belong to the corresponding granule, i.e., instances 3 and 4.

We continue with rules obtained from the DRSA. We now assume that data are ordinal, i.e., there exists a preorder relation D_q on every attribute $q \in Q$. That induces preorder relation D on U as explained in Section 2.1. We recall the dominating set $D^+(u) = \{v \in U : (v,u) \in D\}$, which plays the role of a DRSA granule.

Using the DRSA, we can approximate upward and downward unions, which are sets of instances having at least, resp. at most, a particular value of the decision attribute. By the granular representation, their lower and upper approximations are unions of granules. Using the simple property that $(u,v) \in D \implies D^+(v) \subseteq D^+(u)$, we can eliminate redundant granules (those contained in a bigger granule) and reduce the number of granules covering lower and upper approximations. The rules which can be obtained in this case are called monotonic rules, which have the following form for upward unions:

$$\text{IF att}_1 \geq \text{val}_1 \text{ AND ... AND att}_m \geq \text{val}_m \text{ THEN decision is dec.}$$

Here $\text{val}_1, \ldots, \text{val}_m$ are obtained from the attribute values of the generating instance of that particular granule. Analogously, rules with opposite direction ($\leq$) can be constructed for downward unions.

Again, we provide an example of such a rule from the granule that correspond to instance 3 in Example 5.1.1:

$$\text{IF debt} \leq 3900 \text{ AND salary} \geq 3600 \text{ AND portfolio} \geq 8150$$
$$\text{THEN decision is 1.}$$

Here, attribute “debt" is of cost-type and therefore, the inequality sign is reversed. This is a possible rule which is fulfilled by the instances that belong to the corresponding granule, i.e., instances 3 and 4.

Now we move to the granules that can be induced in the fuzzy IRSA. We recall the T-equivalence relation introduced in Eq. (3.2) and illustrated in Figure 3.1.

In Figure 5.2, we show an example of the granularity of the lower approximation. In this example triangular similarity, the Łukasiewicz t-norm and its R-implicator are used. The top-left sub-figure shows a fuzzy set denoted by the blue line together with its lower approximation, denoted by the green line. The top right sub-figure contains examples of a few granules that can be extracted from the lower approximation. They are represented by red triangles with points on their top. We displayed only seven granules, but in reality every instance generates its own granule. We may see that some granules are included in others

(small triangles inside the bigger ones), so we may safely remove them since they do not contribute to the granular representation of the lower approximation. In the bottom-left subfigure, we see the same example where redundant granules are removed. Sometimes, we want to obtain an even smaller number of granules in order to reduce the number of rules. For example, we may impose the condition that every instance which belongs to the lower approximation to degree at least 0.5, is covered by some granule with degree at least 0.5. In the bottom-right image, we show the reduced set of granules which satisfies this condition.

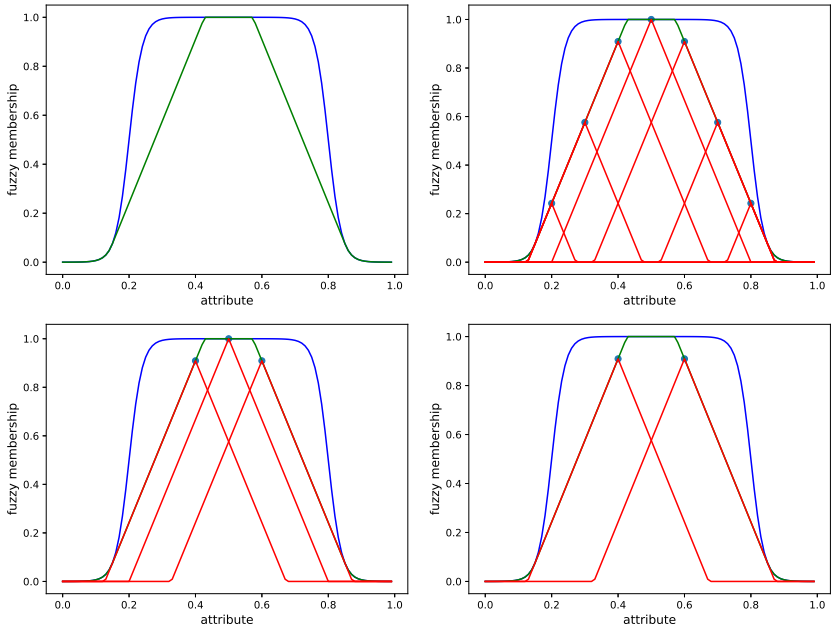

Figure 5.2: Example of lower approximation and its granules

If we assume that we use the fuzzy IRSA on crisp decision classes, we can induce fuzzy rules of the form:

$$\text{IF } \text{att}_1 \sim \text{val}_1 \text{ AND ... AND } \text{att}_m \sim \text{val}_m$$
$$\text{THEN decision is dec with degree at least deg,}$$

where $\sim$ stands for expression “is similar to". Here, as before, $\text{val}_1, \ldots, \text{val}_m$ are obtained from the instance which generates the granule. We now have the degree denoted with “deg”, which represents a membership of the instance in the corresponding decision class. As an example of a such rule, we provide it from the granule generated by instance 3 in Example 5.2.1.

IF debt $\sim$ 3900 AND salary $\sim$ 3600 AND portfolio $\sim$ 8150
THEN decision is 1 with degree at least deg.

Here, degree deg is determined by the membership of a particular instance in the granule. For example, instance 1 fulfills the previous possible rule with degree 0.552. The previous value is obtained from Table 5.1.

The fourth type of rules corresponds to the fuzzy DRSA. We recall the T-preorder (fuzzy dominance) relation from Eq. (3.5) and its visualization given in Figure 3.2. Such a relation consists of two regions: the region of strict dominance and the region of similarity. Hence, the interpretation of rules which correspond to the granules obtained from the fuzzy dominance relation is "greater or similar" ("lower or similar" for the opposite direction). If we assume that we use fuzzy DRSA on crisp upward unions, then the induced fuzzy rules are of the form:

IF $\text{att}_1 \gtrsim \text{val}_1$ AND ... AND $\text{att}_m \gtrsim \text{val}_m$
THEN decision is dec with degree at least deg.

Here, $\gtrsim$ stands for the expression "is greater or similar", and $\text{val}_1, \ldots,$ val_m are obtained from the instance which generates the granule. Again, an example of such rule, we provide it from a granule generated by instance 3 in Example 5.2.1.

IF debt $\lesssim$ 3900 AND salary $\gtrsim$ 3600 AND portfolio $\gtrsim$ 8150
THEN decision is 1 with degree at least deg.

Again, the degree "deg" is determined by the membership of a particular instance in the granule and attribute "debt" is of cost-type and therefore, the inequality is reversed. For example, instance 1 fulfills the previous possible rule with degree 0.647. The previous value is obtained from Table 5.2.

5.4 Granular representation of OWA-based approximations

In practice, data collected for real machine learning problems may be represented as unknown values plus some perturbations. If the amount of perturbations is negligible, we can use the standard fuzzy PRSA approach to obtain the lower and upper approximations. In the opposite case, we require robust methods. As already mentioned in the previous chapter, the OWA-based approach was identified as the most robust

known fuzzy rough approach. OWA-based fuzzy PRSA will yield different lower and upper approximations. However, in general, we cannot claim that the new approximations will not posses inconsistencies and therefore be suitable for the inconsistency correction. In this section, we prove that under for a specific type of fuzzy connectives and for a T-preorder relation, OWA-based fuzzy rough approximations does not posses inconsistencies, i.e., they are granularly representable fuzzy sets.

From Proposition 5.2.1, we already know that a fuzzy set has a granular representation if and only if it is equal to its standard fuzzy rough approximations. Therefore, we should find out under which conditions it holds that:

$$\underline{\mathrm{apr}}_{\widetilde{R}}^{\min,I}(\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)) = \underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A),$$
$$\overline{\mathrm{apr}}_{\widetilde{R}}^{\max,T}(\overline{\mathrm{apr}}_{\widetilde{R}}^{W_U,T}(A)) = \overline{\mathrm{apr}}_{\widetilde{R}}^{W_U,T}(A).$$

To this aim, we recall some definitions and properties about convexity for binary fuzzy logic connectives.

Definition 5.4.1. [6] We say that a binary operator $H : [0,1] \times [0,1] \to [0,1]$ is convex (concave) if for all $x_1, x_2, y_1, y_2 \in [0,1]$ and $w_1, w_2 \in [0,1]$ such that $w_1 + w_2 = 1$, it holds that:

$$H(w_1x_1 + w_2x_2, w_1y_1 + w_2y_2) \leq (\geq)\ w_1H(x_1, y_1) + w_2H(x_2, y_2).$$

Definition 5.4.2. [6] We say that a binary operator $H : [0,1] \times [0,1] \to [0,1]$ is midpoint convex (concave) if for all $x_1, x_2, y_1, y_2 \in [0,1]$, it holds that:

$$H\left(\frac{x_1}{2} + \frac{x_2}{2}, \frac{y_1}{2} + \frac{y_2}{2}\right) \leq (\geq)\ \frac{H(x_1, y_1)}{2} + \frac{H(x_2, y_2)}{2}.$$

Proposition 5.4.1. [6] A continuous midpoint convex (concave) t-norm is convex (concave).

Definition 5.4.3. [95] We say that a binary operator $H : [0,1] \times [0,1] \to [0,1]$ is directionally convex or D-convex (directionally concave or D-concave) if it is a convex (concave) function in both of its arguments, i.e., for all $x_1, x_2, y \in [0,1]$ and $w_1, w_2 \in [0,1]$ such that $w_1 + w_2 = 1$, it holds that:

$$H(w_1x_1 + w_2x_2, y) \leq (\geq)\ w_1H(x_1, y) + w_2H(x_2, y) \quad \text{and}$$
$$H(y, w_1x_1 + w_2x_2) \leq (\geq)\ w_1H(y, x_1) + w_2H(y, x_2).$$

This definition expresses that the partial mappings of H are convex (concave) functions. We prove a simple proposition:

Proposition 5.4.2. Every convex (concave) operator is also D-convex (D-concave).

Proof. Just take $x_1 = x_2 = x$ or $y_1 = y_2 = y$ in the previous definitions. □

The reverse implication is not necessarily satisfied. Now, we formulate and prove the following important result.

Proposition 5.4.3. Let T be a convex left-continuous t-norm and let I be its R-implicator. Then I is concave.

Proof. Assume we are given $x_1, x_2, y_1, y_2, w_1, w_2 \in [0,1]$ such that $w_1 + w_2 = 1$. We have to prove that

$$w_1 I(x_1, y_1) + w_2 I(x_2, y_2) \leq I(w_1 x_1 + w_2 x_2, w_1 y_1 + w_2 y_2).$$

Using the residuation property, we can express this condition as

$$T(w_1 I(x_1, y_1) + w_2 I(x_2, y_2), w_1 x_1 + w_2 x_2) \leq w_1 y_1 + w_2 y_2.$$

By the convexity of T we have that:

$$\begin{aligned} & T(w_1 I(x_1, y_1) + w_2 I(x_2, y_2), w_1 x_1 + w_2 x_2) \\ \leq \; & w_1 T(x_1, I(x_1, y_1)) + w_2 T(x_2, I(x_2, y_2)). \end{aligned}$$

Using the modus ponens property (2.6c), we have that

$$T(x_1, I(x_1, y_1)) \leq y_1 \text{ and } T(x_2, I(x_2, y_2)) \leq y_2.$$

which completes the proof. □

Proposition 5.4.4. Let T be a D-convex left-continuous t-norm and I its R-implicator. Then I is concave in its second argument.

Proof. Similarly as for Proposition 5.4.3. □

We recall the following well-known inequality from calculus which will be needed further on.

Proposition 5.4.5 (Jensen's inequality)**.** [76] Let $f : \mathbb{R} \to \mathbb{R}$ be a convex (concave) function. Let $x_1, \ldots, x_n$ be real numbers and $w_1, \ldots, w_n$ real weights which sum up to 1. Then we have

$$f\left(\sum_{i=1}^{n} w_i x_i\right) \leq (\geq) \sum_{i=1}^{n} w_i f(x_i).$$

The Jensen's inequality holds for arbitrary weighted sum. However, OWA operators will sort the values of the argument set before the weights are multiplied with them. Because of that, we provide the adjusted OWA-Jensen's inequality.

Proposition 5.4.6. Let $f : \mathbb{R} \to \mathbb{R}$ be an increasing and convex (concave) function. Let $x_1, \ldots, x_n$ be real numbers and let W be an OWA vector of weights. Then we have

$$f(\mathrm{OWA}_W(\{x_1, \ldots, x_n\})) \leq (\geq)\ \mathrm{OWA}_W(\{f(x_1), \ldots, f(x_n)\}).$$

Proof. Without loss of generality, we assume that $x_1 \geq \cdots \geq x_n$ and let $W = \{w_1, \ldots, w_n\}$. Then by the Jensen's inequality we have that

$$f(\mathrm{OWA}_W(\{x_1, \ldots, x_n\})) = f(\sum_{i=1}^{n} w_i x_i) \leq (\geq)\ \sum_{i=1}^{n} w_i f(x_i).$$

On the other side, since f is increasing we have that $f(x_1) \geq \cdots \geq f(x_n)$. Therefore, it holds that

$$\sum_{i=1}^{n} w_i f(x_i) = \mathrm{OWA}_W(\{f(x_1), \ldots, f(x_n)\})$$

which completes the proof. □

Before we proceed to the main theorem, we need to address the interchangeability of OWA operators and the min and max operators. We have the following.

Proposition 5.4.7. Let $\{a_{i,j}; i \in I, j \in J\}$ be a matrix of values for I and J being a finite set of indices and let W be an OWA vector of weights. We have that

$$\mathrm{OWA}_W(\{\min\{a_{i,j}; i \in I\}; j \in J\}) \leq \min\{\mathrm{OWA}_W(\{a_{i,j}; j \in J\}); i \in I\}$$

$$\mathrm{OWA}_W(\{\max\{a_{i,j}; i \in I\}; j \in J\}) \geq \max\{\mathrm{OWA}_W(\{a_{i,j}; j \in J\}); i \in I\}$$

Proof. We prove the first expression while the second one holds by analogy. For fixed $i \in I$, we have that $\min\{a_{i,j}; i \in I\} \leq a_{i,j}$ for all $j \in J$. From Proposition 2.4.1, it holds that

$$\mathrm{OWA}_W(\{\min\{a_{i,j}; i \in I\}; j \in J\}) \leq \mathrm{OWA}_W(\{a_{i,j}; j \in J\}),$$

for all $i \in I$. By applying *min* operator on the right side for all $i \in I$, we finish the proof. □

Theorem 5.4.1. Let T be a D-convex left-continuous t-norm and I its R-implicator. Then for every $A \in \mathcal{F}(U)$ it holds that

$$\underline{\mathrm{apr}}_{\widetilde{R}}^{\min,I}(\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)) = \underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A), \quad \overline{\mathrm{apr}}_{\widetilde{R}}^{\max,T}(\overline{\mathrm{apr}}_{\widetilde{R}}^{W_U,T}(A)) = \overline{\mathrm{apr}}_{\widetilde{R}}^{W_U,T}(A).$$

Proof. Observe that using T-transitivity of $\widetilde{R}$ we find

$$\begin{aligned} & \forall w \in U, \widetilde{R}(v,u) \geq T(\widetilde{R}(v,w), \widetilde{R}(w,u)) \\ \Longrightarrow \quad & \widetilde{R}(v,u) \geq \max_{w \in U} T(\widetilde{R}(v,w), \widetilde{R}(w,u)). \end{aligned}$$

Since the equality can be achieved for $w = u$, we can write

$$\widetilde{R}(v,u) = \max_{w \in U} T(\widetilde{R}(v,w), \widetilde{R}(w,u)).$$

First, we provide the proof for the lower approximation. From the inclusion property we know that

$$\underline{\mathrm{apr}}_{\widetilde{R}}^{\min,I}(\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)) \subseteq \underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A).$$

We proceed to prove the opposite inequality. Due to Proposition 5.4.4, I is a concave function in its second argument. Since it is also an increasing function in the second argument, we can apply the OWA-Jensen's inequality from Proposition 5.4.6. We find:

$$\begin{aligned} \underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(u) &= \mathrm{OWA}_{W_L}\left(\left\{I(\widetilde{R}(v,u), A(v)); v \in U\right\}\right) \\ &= \mathrm{OWA}_{W_L}\left(\left\{I(\max_{w \in U} T(\widetilde{R}(v,w), \widetilde{R}(w,u)), A(v)); v \in U\right\}\right) \\ &\leq \mathrm{OWA}_{W_L}\left(\left\{\min_{w \in U} I(T(\widetilde{R}(v,w), \widetilde{R}(w,u)), A(v)); v \in U\right\}\right) \\ &\leq \min_{w \in U} \mathrm{OWA}_{W_L}\left(\left\{I(T(\widetilde{R}(v,w), \widetilde{R}(w,u)), A(v)); v \in U\right\}\right) \\ &= \min_{w \in U} \mathrm{OWA}_{W_L}\left(\left\{I(\widetilde{R}(w,u), I(\widetilde{R}(v,w), A(v))); v \in U\right\}\right) \\ &\leq \min_{w \in U} I\left(\widetilde{R}(w,u), \mathrm{OWA}_{W_L}(\{I(\widetilde{R}(v,w), A(v)); v \in U\})\right) \\ &= \min_{w \in U} I\left(\widetilde{R}(w,u), \underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A)(w)\right) \\ &= \underline{\mathrm{apr}}_{\widetilde{R}}^{\min,I}(\underline{\mathrm{apr}}_{\widetilde{R}}^{W_L,I}(A))(u). \end{aligned}$$

The first inequality follows the monotonicity of I and Proposition 2.4.1. The second inequality is the consequence of Proposition 5.4.7. The third

equality holds from property (2.6f) while the third inequality holds from Proposition 5.4.6.

Next, we provide the proof for the upper approximation. From the inclusion property we have that:

$$\overline{\text{apr}}_{\widetilde{R}}^{\max,T}(\overline{\text{apr}}_{\widetilde{R}}^{W_U,T}(A)) \supseteq \overline{\text{apr}}_{\widetilde{R}}^{W_U,T}(A).$$

We proceed to prove the opposite inequality, using similar arguments:

$$\begin{aligned}
\overline{\text{apr}}_{\widetilde{R}}^{W_U,T}(A)(u) &= \text{OWA}_{W_U}\left(\left\{T(\widetilde{R}(u,v),A(v));\, v\in U\right\}\right)\\
&= \text{OWA}_{W_U}\left(\left\{T(\max_{w\in U} T(\widetilde{R}(u,w),\widetilde{R}(w,v)),A(v));\, v\in U\right\}\right)\\
&\geq \text{OWA}_{W_U}\left(\left\{\max_{w\in U} T(T(\widetilde{R}(u,w),\widetilde{R}(w,v)),A(v));\, v\in U\right\}\right)\\
&\geq \max_{w\in U}\text{OWA}_{W_U}\left(\left\{T(T(\widetilde{R}(u,w),\widetilde{R}(w,v)),A(v));\, v\in U\right\}\right)\\
&= \max_{w\in U}\text{OWA}_{W_U}\left(\left\{T(\widetilde{R}(u,w),T(\widetilde{R}(w,v),A(v)));\, v\in U\right\}\right)\\
&\geq \max_{w\in U} T\left(\widetilde{R}(u,w),\text{OWA}_{W_U}(\{T(\widetilde{R}(w,v),A(v));\, v\in U\})\right)\\
&= \max_{w\in U} T\left(\widetilde{R}(u,w),\overline{\text{apr}}_{\widetilde{R}}^{W_U,T}(A)(w)\right)\\
&= \overline{\text{apr}}_{\widetilde{R}}^{\max,T}(\overline{\text{apr}}_{\widetilde{R}}^{W_U,T}(A))(u).
\end{aligned}$$

□

Therefore, using Corollaries 5.2.2 and 5.2.3, we have the following granular representation of the OWA-based approximations.

$$\begin{aligned}
\underline{\text{apr}}_{\widetilde{R}}^{W_L,I}(A) &= \bigcup\{\widetilde{R}_\lambda^+(u);\ \widetilde{R}_\lambda^+(u)\subseteq \underline{\text{apr}}_{\widetilde{R}}^{W_L,I}(A)\},\\
\overline{\text{apr}}_{\widetilde{R}}^{W_U,T}(A) &= \bigcup\{\widetilde{R}_\lambda^+(u);\ \widetilde{R}_\lambda^+(u)\subseteq \overline{\text{apr}}_{\widetilde{R}}^{W_U,T}(A)\}.
\end{aligned}$$

With this result, we can conclude that OWA-based approximations are granularly representable fuzzy sets under specific conditions. The question remains which connectives preserve the granularity property, or in other words, which left-continuous t-norms are also D-convex.

Example 5.4.1. In this example, we show the data inconsistency correction obtained using the OWA-based fuzzy PRSA and we identify the granules that constitute the OWA-based fuzzy PRSA approximations.

We first calculate the OWA-based fuzzy PRSA approximations of the data from Section 3.2. First, we calculate the lower and upper approximations of the classification dataset from Table 3.2 using T_L-equivalence relation (3.2), IMTL triplet (T_L, I_L, N_L) and exponential OWA weights. As we will see in Section 5.5, T_L is indeed a D-convex t-norm. The relation matrix from Table 3.3 is passed together with the decision attribute to formula (4.3). The obtained lower and upper approximations are given in Table 5.5.

approx. type vs. instance	1	2	3	4	5	6
lower	0.72	0.682	0.414	0.414	0.439	0.442
upper	0.656	0.62	0.7	0.7	0.172	0.209

Table 5.5: The OWA-based fuzzy PRSA in the classification case for the T_L-equivalence relation

In Table 5.6, we present the calculated fuzzy PRSA approximations using T_L-preorder relation (3.5) while the remaining parameters are the same as in Table 5.5.

approx. type vs. instance	1	2	3	4	5	6
lower	0.671	0.298	0.414	0.414	0.288	0.144
upper	0.747	0.62	0.844	0.844	0.362	0.209

Table 5.6: The OWA-based fuzzy PRSA in the classification case for the T_L-preorder relation

We note that the pairs of instances are now indeed consistent. Following the example from Section 3.2, where we identified that instances $u \equiv 6$ and $v \equiv 2$ were inconsistent, using results from Table 5.5, for the lower approximation we obtain $T(\widetilde{R}(u,v), \hat{A}(v)) = T(0.312, 0.682) = 0 \leq 0.442 = \hat{A}(u)$, i.e., they are now consistent. For the upper approximation, we have that $T(\widetilde{R}(u,v), \hat{A}(v)) = T(0.312, 0.62) = 0 \leq 0.209 = \hat{A}(u)$, i.e., we have the consistency again. If we use the results from Table 5.6, for the lower approximation we have that $T(\widetilde{R}(u,v), \hat{A}(v)) = T(0.312, 0.298) = 0 \leq 0.144 = \hat{A}(u)$, i.e., they are consistent. For the upper approximation, we have that $T(\widetilde{R}(u,v), \hat{A}(v)) = T(0.312, 0.62) = 0 \leq 0.209 = \hat{A}(u)$, i.e., we have the consistency again. The values of the fuzzy relations in these examples are obtained from Tables 3.3 and 3.4.

We perform the same calculations for the regression data from Section 3.2 provided in Table 3.5. In order to compute the OWA-based lower and upper approximations w.r.t. T_L-equivalence relation (3.2), we pass the relation values from Table 3.6 and the decision attribute from Table 3.5 formulas (4.3). The obtained granular approximations are given in Table 5.7.

approx. type vs. instance	1	2	3	4	5	6
lower	0.859	0.581	0.731	0.716	0.695	0.597
upper	0.454	0.352	0.492	0.474	0.218	0.395

Table 5.7: The OWA-based fuzzy PRSA in the regression case for the T_L-equivalence relation

In Table 5.8 we calculate the fuzzy PRSA approximations using T_L-preorder relation (3.5) while the other parameters are the same as in Table 5.7.

approx. type vs. instance	1	2	3	4	5	6
lower	0.859	0.556	0.731	0.476	0.69	0.396
upper	0.73	0.559	0.694	0.474	0.358	0.445

Table 5.8: The OWA-based fuzzy PRSA in the regression case for the T_L-preorder relation

We again continue the example from Section 3.2 where we identified that instances $u \equiv 2$ and $v \equiv 3$ are inconsistent. Using values from Table 5.7, for the lower approximation we have that $T(\widetilde{R}(u,v), \hat{A}(v)) = T(0.625, 0.731) = 0.356 \leq 0.581 = \hat{A}(u)$, i.e., they are now consistent. For the upper approximation we have that $T(\widetilde{R}(u,v), \hat{A}(v)) = T(0.625, 0.492) = 0.117 \leq 0.352 = \hat{A}(u)$, i.e., we have the consistency again. If we use the calculated values from Table 5.8, for the lower approximation we have that $T(\widetilde{R}(u,v), \hat{A}(v)) = T(0.571, 0.731) = 0.302 \leq 0.556 = \hat{A}(u)$, while for the upper one we have that $T(\widetilde{R}(u,v), \hat{A}(v)) = T(0.571, 0.694) = 0.265 \leq 0.559 = \hat{A}(u)$. In both cases, we corrected the inconsistency.

We now identify the granules that constitute the OWA-based PRSA approximations of data from Section 3.2 provided in Example 4.2.1. As in Example 5.2.1, in order maintain the readability, we will not include

the granules of the lower approximations. In tables 5.9, 5.10, 5.11 and 5.12, we have the granules that constitute the upper approximations from tables 5.5, 5.6, 5.7, 5.8.

The description is the same as in Example 5.2.1. In every row of each table, the first entry represents the generating instance, while the following 6 entries are the fuzzy membership degrees of every instance from U in the granule generated by the generating instance. For example, In Table 5.9, the first row is a granule generated by instance 1, which is fuzzy set

$$\{(1, 0.656), (2, 0), (3, 0.208), (4, 0.208), (5, 0), (6, 0)\}$$

This granule is obtained in a way that we calculate the T-equivalence values between instance 1 and all other instances where the resulting fuzzy set is:

$$\{(1, 1), (2, 0), (3, 0.552), (4, 0.552), (5, 0), (6, 0)\} \tag{5.4}$$

and then we evaluate T_L operator to the fuzzy degrees from (5.4) and the OWA-based upper approximation membership degree of instance 1 which is 0.656. As in Example 5.2.1, we observe that if we calculate the maximum of every row, we will get the OWA-based upper approximation degrees, i.e., the union of granules is indeed the OWA-based upper approximation.

	1	2	3	4	5	6
1	0.656	0	0.208	0.208	0	0
2	0	0.620	0.032	0.032	0	0
3	0.253	0.112	0.700	0.700	0	0
4	0.253	0.112	0.700	0.700	0	0
5	0	0	0	0	0.172	0
6	0	0	0	0	0	0.209

Table 5.9: Granules of the upper approximation from Table 5.5

	1	2	3	4	5	6
1	0.55	0.216	0.102	0.102	0	0.55
2	0	0.395	0	0	0	0.395
3	0.294	0.647	0.647	0.647	0.449	0.647
4	0.294	0.647	0.647	0.647	0.449	0.647
5	0	0	0	0	0.224	0
6	0	0	0	0	0	0.136

Table 5.10: Granules of the upper approximation from Table 5.6

	1	2	3	4	5	6
1	0.454	0	0	0	0	0
2	0	0.352	0	0	0	0
3	0	0.117	0.492	0	0	0
4	0	0	0	0.474	0	0.166
5	0	0	0	0	0.218	0
6	0	0	0	0.087	0	0.395

Table 5.11: Granules of the upper approximation from Table 5.7

	1	2	3	4	5	6
1	0.730	0.653	0.499	0.730	0.522	0.730
2	0	0.559	0.184	0.476	0	0.559
3	0.122	0.551	0.694	0.694	0.194	0.694
4	0	0	0	0.474	0	0.166
5	0	0	0	0.011	0.358	0
6	0	0	0	0.237	0	0.445

Table 5.12: Granules of the upper approximation from Table 5.8

5.5 Partial characterization of D-convex t-norms

Convexity is a crucial property for the granularity of the OWA-based operators. The general characterization of convex t-norms is still an open problem, but D-convex t-norms with some additional characteristics may be well characterized. The results in this section are mainly an adaptation of the existing work on characterizing convex copulas [83].

Assume that we have a continuous D-convex t-norm T. For a t-norm it is known that it is continuous as a function of two variables, if and only if its partial mappings are continuous [84]. From basic calculus we know that convex functions are continuous on the interior of the domain, which is in this case the open interval $(0,1)$ [118]. However, we may have a discontinuity at the points 0 and 1. An example of a discontinuous D-convex t-norm is the drastic t-norm T_D.

In this section, we want to characterize left-continuous D-convex t-norms. The following proposition shows that such t-norms are necessarily continuous.

Proposition 5.5.1. Every left-continuous D-convex t-norm is continuous.

Proof. As we noted before, a D-convex t-norm can only have discontinuities in 0 or 1. Moreover, a left-continuous t-norm cannot have a discontinuity in 1. Hence, the only possibility is that it is discontinuous in 0. However, we will prove that the partial mappings of any t-norm are right-continuous in 0.

Let $c \in [0,1]$ be a constant. For every $\epsilon > 0$, we need to find $\delta > 0$ such that $x - 0 < \delta \implies T(x,c) - T(0,c) < \epsilon \Leftrightarrow x < \delta \implies T(x,c) < \epsilon$. Taking $\delta = \epsilon/2$ we have that

$$x < \epsilon/2 \implies T(x,c) \leq \min(x,c) \leq \min(\epsilon/2, c) \leq \epsilon/2 < \epsilon,$$

which is true. From this, we conclude there is no discontinuity in 0, i.e., T is continuous in $[0,1]$. □

We proceed with the characterization. First, we show that T cannot have any non-trivial idempotent element. Assume that it has an idempotent point $z \in (0,1)$. From [84], we may then conclude that $T(x,z) = \min(x,z)$ for all $x \in (z,1]$. However, it is easy to see that the function $f(x) = \min(x,c)$ is not convex for any constant $c \in (0,1]$, so T is not a D-convex t-norm. Because of that, we have a contradiction, and T cannot have idempotent points. In particular, the minimum t-norm T_M is not convex.

Under the assumption of continuity, T does not have idempotent points if and only if it is Archimedean [84]. In [83], necessary and sufficient conditions for D-convexity of Archimedean copulas are derived. Here we repeat the proof, adapting it for continuous Archimedean t-norms.

Theorem 5.5.1. Let T be a continuous Archimedean t-norm with a twice differentiable generator f. Then T is D-convex if and only if $1/f'$ is a convex function.

Proof. We recall a representation of T from (2.1):

$$T(x,y) = f^{-1}(\min(f(x)+f(y), f(0))).$$

Since f is twice differentiable, T is D-convex if and only if the conditions $T_{xx}(x,y) \geq 0$ and $T_{yy}(x,y) \geq 0$ hold for all x, y such that $f(x)+f(y) \leq f(0)$, where T_{xx} is the second partial derivative for the first component, while T_{yy} is the second partial derivative for the second one. Due to the symmetry of T, it suffices to show that $T_{xx} \geq 0$. We find:

$$\begin{aligned} & T_{xx}(x,y) \\ = \; & \frac{f''(x)(f'(f^{-1}(f(x)+f(y)))) - f'(x)^2 f''(f^{-1}(f(x)+f(y)))}{(f'(f^{-1}(f(x)+f(y))))^3}. \end{aligned}$$

It holds that $f'(x) < 0$ since f is a strictly decreasing function, so the condition that $T_{xx}(x,y) \geq 0$ is equivalent to

$$\frac{f''(x)}{f'(x)^2} \leq \frac{f''(f^{-1}(f(x)+f(y)))}{f'(f^{-1}(f(x)+f(y)))^2}.$$

We introduce a new variable $u = f^{-1}(f(x)+f(y))$. From the definition we conclude that $u = f^{-1}(f(x)+f(y)) \leq f^{-1}(f(x)) = x$ due to the fact that f^{-1} is also a strictly decreasing function. Now the condition above becomes

$$\frac{f''(x)}{f'(x)^2} \leq \frac{f''(u)}{f'(u)^2}.$$

We have that $\frac{f''(x)}{f'(x)^2} = -(\frac{1}{f'(x)})'$, so the above condition may be rewritten as

$$\left(\frac{1}{f'(x)}\right)' \geq \left(\frac{1}{f'(u)}\right)'. \tag{5.5}$$

Note that for a fixed x, u can take any value smaller than or equal to x. Indeed, from the condition that $f(x)+f(y) \leq f(0)$, it follows that y takes values from the interval $[f^{-1}(f(0)-f(x)), 1]$. Using this as a domain for y, we have that the function $u(y) = f^{-1}(f(x)+f(y))$ is a bijective mapping $[f^{-1}(f(0)-f(x)), 1]$ to $[0, x]$. So for every $u \leq x$, we can choose some y to obtain u.

Since u may take all values smaller or equal to x, we have that the condition (5.5) states that $\left(\frac{1}{f'(x)}\right)'$ is an non-decreasing function. This is further equivalent to $\left(\frac{1}{f'(x)}\right)'' \geq 0$ which means that $\frac{1}{f'(x)}$ is a convex function. □

Example 5.5.1. We present a way to construct a generator satisfying the conditions of the previous theorem. The construction is also inspired by [83] but adapted here to t-norms. Let $g : [0,1] \to [0,\infty]$ be a convex function with $g(1) > 0$. Then the generator can be constructed as:

$$f(x) = \int_x^1 \frac{1}{g(u)} du.$$

By the positivity of g, we ensure that f is a decreasing function, while its convexity ensures that $\frac{1}{f'}$ is a convex function.

To illustrate that our adaptation of the work from [83] brings new knowledge, we need to show that there exists a D-convex t-norm which is not a copula. The following example confirms this.

Example 5.5.2. In Example 5.5.1 take $g(u) = 2 - u$. We have the following:

$$f(x) = \int_x^1 \frac{1}{2-u} du = -\log(2-1) + \log(2-x) = \log(2-x),$$

while $f^{-1}(x) = 2 - e^x$. Using such generator, we construct the associated t-norm:

$$\begin{aligned} T(x,y) &= 2 - e^{\min(\log(2-x)+\log(2-y),\log(2))} = 2 - e^{\log(\min((2-x)(2-y),2))} \\ &= 2 - \min((2-x)(2-y),2) = \max(2(x+y-1)-xy,0). \end{aligned}$$

If we take values $x = 0.5, y = 0.9, x' = 0.4, y' = 0.8$, we can see that the 2-increasingness property does not hold, i.e.

$$T(x,y) + T(x',y') < T(x',y) + T(x,y').$$

which means that T is not a copula.

Furthermore, we can easily check, with the same values, that T is not midpoint convex, which is equivalent to stating that it is not convex.

Example 5.5.3. We check the D-convexity of the left-continuous t-norms from Table 2.1.

- Łukasiewicz t-norm $T_L(x,y) = \max(x+y-1, 0)$ is D-convex since its partial mappings are a composition of a linear function and max, which are both convex. It was proven in [7] that T_L is even convex.
- Product t-norm $T_P(x,y) = xy$ is D-convex because its partial mappings are linear functions.
- From the above exposition, we already know that the minimum t-norm $T_M(x,y) = \min(x,y)$ is not D-convex.
- The nilpotent minimum t-norm

$$T_{nM}(x,y) = \begin{cases} \min(x,y) & \text{if } x+y > 1, \\ 0 & \text{otherwise.} \end{cases}$$

  is not D-convex because its partial mappings have discontinuities in the interior $(0,1)$ of its domain.

5.6 Conclusion

In this chapter, we analysed the granular representability of crisp and fuzzy sets w.r.t. a (fuzzy) preorder relation. We introduced the notion of a granularly representable (fuzzy) set as a union of simple granules, where granules represent the fuzzy equivalence or dominance classes of individual instances. As our main contribution, we proved that OWA-based fuzzy rough approximations are granularly representable fuzzy sets when we use D-convex left-continuous t-norms and their residual implicators for calculation of the approximations. Finally, we characterized continuous convex t-norms and we presented a method to construct them.

Chapter 6

Granular Approximations: a Statistical Learning Approach for Inconsistency Handling

This chapter is motivated by the Kotłowski-Słowiński (KS) approach [89] described in Sections 1.3 and 2.3, in the sense that we generalize the monotonicity constraints using fuzzy relations while the ordinal classes are replaced with fuzzy membership degrees. Instead of a crisp preorder relation (or an equivalence relation if it is symmetric), we will now consider a fuzzy T-preorder relation to model the relationship between different instances on the condition attributes, where T refers to a given left-continuous t-norm that models conjunction in fuzzy logic. The T-preorder relations also include T-equivalence relations that can measure (symmetric) similarity between numerical vectors. Moreover, the new approach requires that the decision attribute is a fuzzy set, i.e., it has to take values from interval $[0,1]$. Hence, it is appropriate for problems where the decision attribute can be modeled using values from this interval; as we will explain, this is the case for binary classification and regression problems.

Just like the KS approach, our proposal is also interesting from the granular computing point of view. In particular, the sets obtained with the KS approach, as well as with the novel approach, can be represented as unions of meaningful simple sets, i.e., they are granularly representable [63, 138]. Due to the granular properties of our new approach, we call its result a *granular approximation*.

The remainder of the chapter is organized as follows. In Section 6.1,

we develop the statistical foundations of granular approximations. Section 6.2 deals with optimization problems that output granular approximations, while some of their important properties are proven in Section 6.3. In Section 6.4 we deal with the dual formulations of the optimization problems introduced in Section 6.2. Using the duality theory, we obtain greedy algorithms for the optimization problems from Section 6.2 that allow us to prove Proposition 6.3.3. Section 6.5 is reserved for the conclusion.

6.1 Statistical approach to inconsistency in data

6.1.1 Ontic fuzzy sets and probabilistic uncertainty

Before we proceed with the statistical view on the inconsistency in data, we have to distinguish between probabilistic uncertainty and fuzziness since both will be used in the development of the approach. Fuzzy sets are often related to uncertainty modeling [80, 102, 31]. However, we should be very careful when mentioning that fuzzy sets are used to model uncertainty. First, two types of classical (crisp) sets have to be distinguished: conjunctive and disjunctive sets [135]. A conjunctive set is a collection of items that represents a well known complex entity, i.e., it is a conjunction of its elements. For example, a time interval that describes a span of some activity. On the other hand, a disjunctive set describes incomplete information about an ill-known object. The object of interest is contained in the disjunctive set but we do not know which element it is, i.e., the set is a disjunction of its elements. For example, an event that occurred at an unknown moment in time is described with a time interval that represents our knowledge about the unknown event. Conjunctive sets are also known as ontic sets while disjunctive sets are called epistemic sets. Fuzzy sets are used to model gradual information which is not uncertain by itself. Fuzzy sets may be related to uncertainty only if the underlying universe, on which a fuzzy set is defined, is a disjunctive set. In that case, fuzzy sets make incomplete knowledge more expressive by allowing gradual information. Such fuzzy sets are usually known as epistemic fuzzy sets and form the basis of possibility theory [147]. In the remainder of the thesis, we always use fuzzy sets defined over a conjunctive universe, i.e., ontic fuzzy sets, while we assume that the uncertainty in data is of probabilistic nature and it is soley related to the uknown membership degrees. An example of an ontic fuzzy set is a set of apartments that are “expensive", i.e., a fuzzy set whose universe is

some set of apartments and its membership function is a price measure of those apartments. The price is an actual economical characteristic. In such settings, no uncertainty or lack of knowledge exists about the set of apartments which is a conjunctive set. The uncertainty we assume exists around actual prices of apartments or "degrees of expensiveness", and such uncertainty will be modeled using probability distributions.

6.1.2 Granularly representable random fuzzy sets

We assume that we observed a finite set of instances U from the underlying universe, i.e., U is a random sample. U is described with condition and decision attributes where the decision attribute takes values in [0,1], which are interpreted as membership degrees to an unknown fuzzy set that we want to reconstruct using the observed values. From the perspective of statistical learning theory introduced in Subsection 2.3.1, condition attributes correspond to random variable $\mathcal{X}$ while the decision attribute corresponds to random variable $\mathcal{Y}$, which now takes values from interval $[0,1]$. The fuzzy set that we want to reconstruct contains uncertainties that are represented in a probabilistic way, i.e., we assume that the actual values are altered due to perturbation. Perturbation may be caused by incomplete knowledge about data (missing attributes) or by random effects that occur during data generation. Such altered values are represented by a family of random variables $\{\mathcal{A}(u), u \in U\}$ which model our uncertainty about the ill-known membership degrees $\{A(u), u \in U\}$. In other words, for each instance u, the ill-known membership degree $A(u)$ is represented with the random variable $\mathcal{A}(u)$ having codomain $[0,1]$. Family $\{\mathcal{A}(u), u \in U\}$ is a special case of a *random fuzzy set* defined in [107] (the other name is *fuzzy random variable*). Hence, we may refer to the family as *random fuzzy set* $\mathcal{A}$.

The family $\{\mathcal{A}(u), u \in U\}$ corresponds to family $\mathcal{Y}_{\mathcal{X}=x}$ from Subsection 2.3.1. Therefore, we formulate the reconstruction of fuzzy set A as the problem where for a given set of instances U and its condition and decision attributes, we want to estimate characteristics of $\mathcal{A}(u)$ (like conditional mean, median and quantiles mentioned above) in order to describe the ill-known $A(u)$. Knowledge about condition attributes is represented using a T-preorder relation $\widetilde{R}$, i.e., for each pair $u, v \in U$ we are given the value $\widetilde{R}(u,v)$. We denote the observed decision values as $\bar{A}(u)$ for $u \in U$.

In the first step, we will extend the probabilistic monotonicity constraints (2.13) for a T-preorder relation. In order to relate granular rep-

resentability and the family of random variables $\{\mathcal{A}(u), u \in U\}$, we introduce the following definition.

Definition 6.1.1. Random fuzzy set $\mathcal{A}$ is granularly representable (does not posses inconsistencies) if

$$\forall u, v \in U \text{ and } \forall p \in [0,1], \widetilde{R}(u,v) \leq I(A_p(v), A_p(u)),$$

where $A_p(u) = Q_{\mathcal{A}(u)}(p)$, i.e., A_p is the conditional p-quantile of $\mathcal{A}$.

Definition 6.1.1 is an extension of the third equivalence in (2.13). It states that $\mathcal{A}$ is granularly representable if all its p-quantiles A_p ($p \in [0,1]$) are granularly representable as ordinary fuzzy sets.

The next question is, if the random fuzzy set $\mathcal{A}$ is granularly representable, is its expected value $E\mathcal{A}$, defined as $E\mathcal{A} = \{E(\mathcal{A}(u)), u \in U\}$, also granularly representable? Before answering this question, we recall the well-known Jensen inequality [117].

Proposition 6.1.1. Let μ be a probability measure on the set of reals, g a μ-measurable real function and ϕ a real convex function. It holds that

$$\int \phi(g) d\mu \geq \phi\left(\int g d\mu \right).$$

Since the standard (Lebesgue) measure is equivalent to the probability measure on $[0,1]$ (measure value of interval $[0,1]$ is 1), the above inequality translates to

$$\int_0^1 \phi(g(x)) dx \geq \phi\left(\int_0^1 g(x) dx \right).$$

Using Jensen's inequality, we obtain the following result.

Proposition 6.1.2. Let T be a D-convex t-norm and I its R-implicator. Then $E\mathcal{A}$ is granularly representable (does not posses inconsistencies) as soon as $\mathcal{A}$ is.

Proof. For every $u, v \in U$, we need to prove that

$$T(\widetilde{R}(u,v), E\mathcal{A}(v)) \leq E\mathcal{A}(u).$$

Using (2.9), we have that $\forall u \in U$, $E\mathcal{A}(u) = \int_0^1 A_p(u) dp$. It follows that

$$T(\widetilde{R}(u,v), E\mathcal{A}(v)) = T\left(\widetilde{R}(u,v), \int_0^1 A_p(v) dp \right)$$

$$\leq \int_0^1 T(\widetilde{R}(u,v), A_p(v))dp$$
$$\leq \int_0^1 A_p(u)dp = E\mathcal{A}(u).$$

The first inequality follows from the fact that $T(c,\cdot)$ is a convex function for a constant c and Jensen's inequality. The second inequality follows from the granularity of A_p. □

6.2 Calculation of granular approximations

In this section, we discuss which properties of $\mathcal{A}$ can be estimated and how to do this in practice. In general, the observed fuzzy set $\bar{A}$ is not granularly representable due to the presence of inconsistency, so our goal is to find a granularly representable set that is close to it by minimizing a certain empirical risk. For given loss function L, the general form of the optimization problem expressing our goal is

$$\begin{aligned} \text{minimize} \quad & \sum_{u\in U} L(\bar{A}(u), \hat{A}(u)) \\ \text{subject to} \quad & T(\widetilde{R}(u,v), \hat{A}(v)) \leq \hat{A}(u), \quad u,v \in U \\ & 0 \leq \hat{A}(u) \leq 1, \quad u \in U, \end{aligned} \tag{6.1}$$

where $\{\hat{A}(u), u \in U\}$ is the unknown granularly representable set. We will call the result of optimization problem (6.1) the *granular approximation* of fuzzy set $\{\bar{A}(u), u \in U\}$.

Optimization problem (6.1) is the main contribution of the chapter. It allows us to remove inconsistencies (obtain a granularly representable set) with the least cost of alteration of values (w.r.t. loss function L). The remainder of the section investigates specific cases for which problem (6.1) can be efficiently solved.

Under the assumption that $\mathcal{A}$ is granularly representable, it is desirable to use loss functions for which the Bayes predictor is granularly representable as well.

Definition 6.2.1. We say that a loss function L is *granular* with respect to a left-continuous t-norm T and T-preorder $\widetilde{R}$ if its Bayes predictor is granularly representable under the assumption that the underlying family of random variables $\{\mathcal{A}(u), u \in U\}$ is granularly representable w.r.t. T and $\widetilde{R}$.

Note that with this definition, the p-quantile loss function (2.14) is granular, since its Bayes predictor is the quantile fuzzy set A_p, which is granularly representable by the definition. The squared error loss function (2.10) is granular for D-convex t-norms since the Bayes predictor $E\mathcal{A}$ is granularly representable in this case by Proposition 6.1.2. Hence, both loss functions introduced in Subsection 2.3.1 are suitable for the calculation of granular approximations.

In problem (6.1), both objective function and constraints are not necessarily linear and may take different forms that depend on loss function L and on the type of fuzzy logic connectives used. However, in case of the loss functions (2.14) and (2.10), and continuous Archimedean t-norms, the optimization problem can be efficiently solved.

Indeed, consider t-norms $T_{L,\varphi}$ and $T_{P,\varphi}$ introduced in Eq. (2.2) and (2.3). If $T_{L,\varphi}$ is used in (6.1), then the set of constraints that express granular representability can be simplified in the following way: for all $u, v \in U$,

$$\begin{aligned}
& \widetilde{R}(u,v) \leq I_{L,\varphi}(\hat{A}(v), \hat{A}(u)) \\
\Leftrightarrow \quad & T_{L,\varphi}\left(\widetilde{R}(u,v), \hat{A}(v)\right) \leq \hat{A}(u) \\
\Leftrightarrow \quad & \varphi^{-1}(\max(\varphi(\widetilde{R}(u,v)) + \varphi(\hat{A}(v)) - 1, 0) \leq \hat{A}(u) \\
\Leftrightarrow \quad & \max(\varphi(\widetilde{R}(u,v)) + \varphi(\hat{A}(v)) - 1, 0) \leq \varphi(\hat{A}(u)) \\
\Leftrightarrow \quad & \max(\widetilde{R}_\varphi(u,v) + \alpha_v - 1, 0) \leq \alpha_u \\
\Leftrightarrow \quad & \widetilde{R}_\varphi(u,v) \leq \alpha_u - \alpha_v + 1
\end{aligned}$$

where we introduced the shorthands $\widetilde{R}_\varphi(u,v) = \varphi(\widetilde{R}(u,v))$, $\alpha_u = \varphi(\hat{A}(u))$ and $\alpha_v = \varphi(\hat{A}(v))$. The last equivalence holds because 0 is always smaller than α_u, hence the max operator can be lifted.

If $T_{P,\varphi}$ is used then in an analogous way we find

$$\varphi^{-1}(\varphi(\widetilde{R}(u,v))\varphi(\hat{A}(v))) \leq \hat{A}(u) \Leftrightarrow \alpha_v \widetilde{R}_\varphi(u,v) \leq \alpha_u$$

for all $u, v \in U$.

The border constraints now become $0 \leq \alpha_u \leq 1$ for all $u \in U$. We can conclude that using continuous Archimedean t-norms leads to linear optimization constraints. This is a promising result since many optimization solvers are very efficient with linear constraints.

In both cases, the empirical risk can be expressed as

$$\sum_{u \in U} L(\bar{A}(u), \varphi^{-1}(\alpha_u)).$$

In the empirical risk above, the non-linear term $\varphi^{-1}(\alpha_u)$ appears. Function φ^{-1} is an arbitrary bijection which can lead to a non-convex optimization problem. However, Proposition 2.3.2 states that a different scaling of values does not change the Bayes predictor delivered by the p-quantile loss function. To eliminate the non-linearity, we can use the scaled loss $L_{p,\varphi}(\bar{A}(u), \varphi^{-1}(\alpha_u)) = L_p(\varphi(\bar{A}(u)), \alpha_u)$ instead of $L_p(\bar{A}(u), \varphi^{-1}(\alpha_u))$. Although the value of the estimand (the quantity that is estimated, i.e., the Bayes predictor A_p) remains unchanged with the scaled loss function, the estimator (the result of the optimization $\hat{A}_p$) can be different. From the theory of quantile regression, we can express the optimization of the mean pinball loss as a linear program [85]. We introduce new variables $x_u, u \in U$ and $y_u, u \in U$ such that $x_u = \max(\varphi(\bar{A}(u) - \alpha_u), 0)$, $y_u = \max(\alpha_u - \varphi(\bar{A}(u)), 0)$, as well as the shorthand $\bar{A}_\varphi(u) = \varphi(\bar{A}(u))$. In case $T_{L,\varphi}$ is used, we can reformulate optimization problem (6.1) as

$$\begin{aligned} \text{minimize} \quad & p \sum_{u \in U} x_u + (1-p) \sum_{u \in U} y_u, & \\ \text{subject to} \quad & \alpha_u - \alpha_v + 1 \geq \widetilde{R}_\varphi(u, v), & u, v \in U \\ & x_u - y_u = \bar{A}_\varphi(u) - \alpha_u, & u \in U \\ & 0 \leq \alpha_u \leq 1, x_u \geq 0, y_u \geq 0. & u \in U \end{aligned} \tag{6.2}$$

In case of $T_{P,\varphi}$, optimization problem (6.1) obtains the form

$$\begin{aligned} \text{minimize} \quad & p \sum_{u \in U} x_u + (1-p) \sum_{u \in U} y_u, & \\ \text{subject to} \quad & \alpha_v \widetilde{R}_\varphi(u, v) \leq \alpha_u, & u, v \in U \\ & x_u - y_u = \bar{A}_\varphi(u) - \alpha_u, & u \in U \\ & 0 \leq \alpha_u \leq 1, x_u \geq 0, y_u \geq 0. & u \in U \end{aligned} \tag{6.3}$$

Summarizing, for mean pinball loss and a continuous Archimedean t-norm, the optimization problem (6.1) can be expressed as a linear program and, therefore, efficiently solved using one of many existing efficient linear programming solvers. We have the following technical result.

Proposition 6.2.1. Constraints $0 \leq \alpha_u \leq 1, u \in U$ in (6.2) and (6.3), are redundant.

Proof. Assume that the constraints are removed and that an optimal solution $\alpha_u^*, u \in U$, has values smaller than 0 or larger than 1. We construct another solution from $\alpha_u^*, u \in U$, by replacing values larger than

1 by 1, and values smaller than 0 by 0. It is easy to check that the new solution satisfies the consistency constraints. From the constraints $x_u - y_u = \bar{A}_\varphi(u) - \alpha_u, u \in U$, it is easy to see that when $\alpha_u \geq 1$ then $\bar{A}_\varphi(u) - \alpha_u \leq 0$, which leads to $x_u = 0$ and $y_u = \alpha_u - \bar{A}_\varphi(u)$, and when $\alpha_u \leq 0$ then $\bar{A}_\varphi(u) - \alpha_u \geq 0$, which leads to $x_u = \bar{A}_\varphi(u) - \alpha_u$ and $y_u = 0$. Hence, after replacing values larger than 1 by 1, the values of y_u will be reduced and after replacing values smaller than 0 by 0, the values of x_u will also be reduced. In both cases, the value of the objective function will be reduced. Therefore, we constructed a feasible solution with a smaller cost which contradicts the optimality of $\alpha^*_u, u \in U$. □

A solution of linear problems (6.2) and (6.3) is not necessarily unique as a consequence of linearity of both objective function and constraints. However, if for some probability parameter p we have infinitely many solutions, the lower and upper bounds of such family of solutions can be calculated by running the linear programs with parameters $p - \epsilon$ and $p + \epsilon$, respectively, for sufficiently small ϵ.

If the squared error loss is used as a loss function, it is obvious that the objective function will become non-linear. Also, Proposition 2.3.2 does not hold anymore and using scaled loss $L_{p,\varphi}(\bar{A}(u), \varphi^{-1}(\alpha_u)) = L_p(\bar{A}_\varphi(u), \alpha_u)$ instead of $L_p(\bar{A}(u), \varphi^{-1}(\alpha_u))$ will lead to the estimation of a different Bayes predictor. However, we will include this approach in our analysis since it may give good results in practical applications. In this case, the optimization problem for the t-norm $T_{L,\varphi}$ is

$$\begin{aligned} \text{minimize} \quad & \sum_{u \in U} (\alpha_u - \bar{A}_\varphi(u))^2, \\ \text{subject to} \quad & \alpha_u - \alpha_v + 1 \geq \widetilde{R}_\varphi(u, v), \quad u, v \in U \\ & 0 \leq \alpha_u \leq 1, \qquad\qquad\quad u \in U \end{aligned} \tag{6.4}$$

while for $T_{P,\varphi}$ the corresponding problem is

$$\begin{aligned} \text{minimize} \quad & \sum_{u \in U} (\alpha_u - \bar{A}_\varphi(u))^2, \\ \text{subject to} \quad & \alpha_v \widetilde{R}_\varphi(u, v) \leq \alpha_u, \qquad u, v \in U \\ & 0 \leq \alpha_u \leq 1. \qquad\qquad u \in U \end{aligned} \tag{6.5}$$

Using a similar argument as in Proposition 6.2.1, we may drop the constraints $0 \leq \alpha_u \leq 1, u \in U$.

To solve the proposed linear and quadratic programs, we have two approaches: geometrical or combinatorial. The combinatorial approach

for the linear programs is discussed in Section 6.4. Namely, the dual versions of problems (6.2) and (6.3) can be modeled as the min-cost flow problem and its variations. We recall the min-cost flow problem and some algorithms used to solve it in 6.4.1 while we show how to model dual problems of (6.2) and (6.3) as the min-cost flow problem and a variation of the min-cost flow problem, respectively, in 6.4.2. In the same section, we provide a greedy algorithm to solve the aforementioned variation based on the successive shortest path algorithm that solves the original min-cost flow problem. Since the algorithm is new, we provide its proof of correctness in 6.4.3. The combinatorial approach, i.e., the duality of the quadratic programs were not discussed. Regarding the time complexity of the combinatorial optimization approach, The successive shortest path algorithm posses a pseudo-polynomial complexity, which depends on the constant we multiply with p and $\bar{A}$ to make them integers, which is important in the development of the algorithm [1]. However, there are also polynomial algorithms for solving the min-cost flow problem like repeated capacity scaling algorithm with complexity $O(|U|^6 \log(|U|))$ and enhanced capacity scaling algorithm with complexity $O(|U|^4 \log(|U|))$ [2].

The geometrical approach includes the aforementioned simplex methods. They are based on geometrical structures that are created in space by the constraints and the objective function. There are many softwares that are able to solve linear and quadrtic programs like Gurobi [68] and Mosek [9]. We need to bear in mind that the proposed optimization problems have $O(|U|)$ variables and $O(|U|^2)$ constraints which leads to the constraint matrix with $O(|U|^3)$ entries. For a large sample size, dealing with such matrix can be computationally demanding. However, the matrix is sparse (vast majority of entries are 0) and our internal experiments showed that Mosek solver can be used as a efficient option to deal with this sparse constraint matrix.

Example 6.2.1. This example continues with the data introduced in Section 3.2. We want to calculate granular approximations using optimization procedures (6.2) and (6.4) that are developed for the Łukasiewicz t-norm T_L.

First, we calculate the granular approximation of the classification dataset from Table 3.2 using T_L-equivalence relation (3.2) and quantile loss L_p. The relation matrix from Table 3.3 is passed together with the decision attribute to the optimization problem (6.2) with probability parameters $p \in \{0, 0.25, 0.5, 0.75, 1\}$. The obtained granular approximations are given in Table 6.1.

p vs. instance	1	2	3	4	5	6
0.000	0.448	0.588	0.000	0.000	0.000	0.000
0.250	0.448	0.588	0.000	0.000	0.000	0.000
0.500	1.000	0.687	0.552	0.552	0.000	0.000
0.750	1.000	1.000	1.000	1.000	0.235	0.313
1.000	1.000	1.000	1.000	1.000	0.235	0.313

Table 6.1: Granular approximations in the classification case for the p-quantile loss and T-equivalence relation

In Table 6.2, we present the calculated granular approximations using T_L-preorder relation (3.5) while the remaining parameters are the same as in Table 6.1.

p vs. instance	1	2	3	4	5	6
0.000	0.353	0.000	0.000	0.000	0.000	0.000
0.250	0.743	0.390	0.390	0.390	0.000	0.000
0.500	1.000	0.390	0.793	0.793	0.000	0.000
0.750	1.000	1.000	1.000	1.000	0.610	0.313
1.000	1.000	1.000	1.000	1.000	0.610	0.313

Table 6.2: Granular approximations in the classification case for the p-quantile loss and T-preorder relation

The interpretation of both tables is analogous. In every row, we have a granular approximation for a corresponding probability parameter from the first column. Every entry is a fuzzy membership degree for the corresponding instance which may be interpreted as the degree up to which the instance belongs to class with label 1. Since that fuzzy value is unknown, we have its distribution characterized with quantiles. For example, in the second row of Table 6.1, we say that with probability 0.25, the degree up to which instance 3 belongs to the class with label 1 is not greater than 0.588, while in the case of Table 6.2, the degree is not greater than 0.390.

The granular approximations obtained using optimization problem (6.4) and T_L-equivalence relation (3.2) are shown in Table 6.3, while the output of the same optimization procedure using T_L-preorder relation (3.5) is provided in Table 6.4.

instance	1	2	3	4	5	6
degree	0.965	0.817	0.517	0.517	0.053	0.130

Table 6.3: Granular approximations in the classification case for the squared error loss and T-equivalence relation

instance	1	2	3	4	5	6
degree	0.960	0.607	0.607	0.607	0.217	0.000078

Table 6.4: Granular approximations in the classification case for the squared error loss and T-preorder relation

In this case, we may say that the expected degree to which instance 3 belongs to the class with label 1 is equal to 0.517 in the case of the T_L-equivalence, and it is equal to 0.607 in the case of the T_L-preorder.

We note that the pairs of instances are now indeed consistent. Following the example from Section 3.2, where we identified that instances $u \equiv 6$ and $v \equiv 2$ were inconsistent, using results from Table 6.3, we obtain $T(\widetilde{R}(u,v)\hat{A}(v)) = T(0.312, 0.817) = 0.129 \leq 0.13 = \hat{A}(u)$, i.e., they are now consistent. If we use the results from Table 6.4, we have that $T(0.312, 0.607) = 0 \leq 0.000078 = \hat{A}(u)$, i.e., they are consistent. The values of the fuzzy relations in these examples are obtained from Tables 3.3 and 3.4.

We perform the same calculations for the regression data from Section 3.2 provided in Table 3.5. In order to compute the granular approximations w.r.t. quantile loss and T_L-equivalence relation (3.2), we pass the relation values from Table 3.6 and the decision attribute from Table 3.5 to optimization procedure (6.2) with probability parameters $p \in \{0, 0.25, 0.5, 0.75, 1\}$. The obtained granular approximations are given in Table 6.5.

p vs. instance	1	2	3	4	5	6
0.000	0.770	0.240	0.615	0.608	0.400	0.300
0.250	0.770	0.240	0.615	0.608	0.400	0.300
0.500	0.770	0.425	0.800	0.608	0.400	0.300
0.750	0.770	0.445	0.820	0.850	0.400	0.542
1.000	0.770	0.445	0.820	0.850	0.400	0.542

Table 6.5: Granular approximations in the regression case for the p-quantile loss and T-equivalence relation

In Table 6.6 we calculate the granular approximations using T_L-preorder relation (3.5) while the other parameters are the same as in Table 6.5.

p vs. instance	1	2	3	4	5	6
0.000	0.770	0.240	0.615	0.323	0.400	0.240
0.250	0.770	0.300	0.675	0.383	0.400	0.300
0.500	0.770	0.425	0.800	0.508	0.400	0.300
0.750	0.770	0.663	0.820	0.746	0.400	0.538
1.000	0.850	0.767	0.850	0.850	0.504	0.642

Table 6.6: Granular approximations in the regression case for the p-quantile loss and T-preorder relation

The obtained fuzzy values are estimations of quantiles of the expensiveness, under the assumption that it is a random fuzzy set and that its realizations are given in Table 3.5. We interpret the values in a way that, for example, in the third row of Table 6.5 we say that the expensiveness of instance 2 is less than 0.24 with probability 0.25, or in the fourth row of the table, we say that the expensiveness of instance 4 is less than 0.85 with probability 0.75. In the case of Table 6.6 we say that the expensiveness of instance 2 is less than 0.3 with probability 0.25, or in the fourth row of the table, we say that the expensiveness of instance 4 is less than 0.746 with probability 0.75.

The results for the squared error loss used in optimization procedure (6.4) are shown in Tables 6.7 and 6.8 for T_L-equivalence (3.2) and T_L-preorder (3.5).

instance	1	2	3	4	5	6
degree	0.770	0.343	0.718	0.729	0.400	0.421

Table 6.7: Granular approximations in the classification case for the squared error loss and T-equivalence relation

instance	1	2	3	4	5	6
degree	0.770	0.477	0.820	0.560	0.400	0.352

Table 6.8: Granular approximations in the classification case for the squared error loss and T-preorder relation

In the case of Table 6.7, we say that the expected expensiveness of instance 4 is equal to 0.729, while in the case of Table 6.8 the expected expensiveness of instance 4 is equal to 0.56.

We again continue the example from Section 3.2 where we identified that instances $u \equiv 2$ and $v \equiv 3$ are inconsistent. Using estimated values from Table 6.7 we have that $T(\widetilde{R}(u,v), \hat{A}(v)) = T(0.625, 0.718) = 0.343 \leq 0.343 = \hat{A}(u)$, i.e., they are now consistent. Also, using estimated values from Table 6.8 we have that $T(\widetilde{R}(u,v), \hat{A}(v)) = T(0.571, 0.82) = 0.391 \leq 0.477 = \hat{A}(u)$, i.e., they are also consistent in this case.

Throughout this example, we note that all the estimations and results we obtained depend on the fuzzy relation that is used, i.e., whether it is a similarity or a fuzzy dominance relation. The choice of such relation will depend on the meaning of the particular dataset and the decision by the creator of the model whether similarity or fuzzy dominance (or some other fuzzy relation) is more appropriate to describe the relationship between instances.

6.3 Properties

In this section, we prove some properties of the granular approximations obtained in Section 6.2. The first two propositions show that the proposed approach is indeed a generalization of both the KS approach for the binary classification case, and of the standard fuzzy rough set approximations.

Proposition 6.3.1. If $\widetilde{R}$ and $\bar{A}$ are crisp, then Problem (6.1) is reduced to Problem (2.16) for $K = 2$.

Proof. If $\bar{A}$ is crisp, it is obvious that the objective function from (6.1) corresponds to the objective function from (2.16) for $K = 2$, where the labels with value 1 are those that are more preferred. Regarding the constraints, we examine the consistency conditions in form $\widetilde{R}(u,v) \leq I(\hat{A}(v), \hat{A}(u))$. If $\widetilde{R}(u,v) = 0$, then there are no restrictions on the implication, i.e., we do not have a constraint. If $\widetilde{R}(u,v) = 1$ then $\hat{A}(v) \leq \hat{A}(u)$ from the ordering property of I (2.6d). Since $\widetilde{R}(u,v) = 1$ means that $u \succeq v$ (u dominates v) then the condition $\widetilde{R}(u,v) = 1 \Rightarrow \hat{A}(u) \geq \hat{A}(v)$ is equivalent to $u \succeq v \Rightarrow \hat{A}(u) \geq \hat{A}(v)$ which is exactly the condition from (2.16). □

Proposition 6.3.2. The respective lower fuzzy rough approximations are solutions of the optimization problems (6.2) and (6.3) for probability parameter $p = 0$, while the respective upper fuzzy rough approximations are solutions of the same problems for probability parameter $p = 1$.

Proof. When optimization problems (6.2) and (6.3) are considered in terms of α and not in terms of $\hat{A}$, they can be seen as problem (6.1) with t-norm T_L or T_P, relation $\widetilde{R}_\varphi$ and observations $\bar{A}_\varphi$. If $p = 1$, then the loss function for $u \in U$ is equal 0 if $\alpha_u - \bar{A}_\varphi(u) \geq 0$ and to a positive value otherwise. If for all $u \in U$ it holds that $\alpha_u \geq \bar{A}_\varphi(u)$, then the objective is 0 and hence any such α is a solution. Such fuzzy set α contains fuzzy set $\bar{A}_\varphi$ and is granularly representable w.r.t. t-norm T_L or T_P and relation $\widetilde{R}_\varphi$. From Proposition 5.2.2, the smallest such α is the fuzzy rough upper approximation, i.e., the smallest solution is

$$\alpha^*_u = \max_{v \in U} T_L(\widetilde{R}_\varphi(v,u), \bar{A}_\varphi(v)),$$

or with T_P instead of T_L. Then, the final solution $\hat{A}^*$ is obtained

$$\begin{aligned}
\hat{A}^*(u) &= \varphi^{-1}(\alpha^*_u) \\
&= \varphi^{-1}(\max_{v \in U} T_L(\widetilde{R}_\varphi(v,u), \bar{A}_\varphi(v))) \\
&= \max_{v \in U} \varphi^{-1}(T_L(\varphi(\widetilde{R}(v,u)), \varphi(\bar{A}(v)))) \\
&= \max_{v \in U} T_{L,\varphi}(\widetilde{R}(v,u), \bar{A}(v)) = \overline{\mathrm{apr}}^{\max, T_{L,\varphi}}_{\widetilde{R}}(A)(u).
\end{aligned}$$

The derivation for T_P is the same.

The proof for the lower approximation is analogous. □

We examine Proposition 6.3.2 from the perspective of knowledge representation. The lower and upper fuzzy rough approximations are

seen as sets of necessary and possible knowledge respectively. In other words, the actual ill-known knowledge must contain the lower approximation and be contained in the upper one. In probabilistic terms, the probability that the actual knowledge is between these approximations is 1. Hence, the lower and upper approximations are the extreme values in the probability distributions of the actual knowledge. It means that the lower approximation is the 0-quantile while the upper approximation is the 1-quantile.

The inconsistency correction performed by rough set approximations can be considered as extreme, since the resulting approximations are either a subset (lower approximation) or a superset (upper approximation) of the original (fuzzy) set. It is this an interesting question if a family of approximations that lie in between lower and upper approximations can be constructed in a way that there exists a monotonic ordering of them. The ordering is motivated by the fact that the lower approximation is always a subset of the upper one. The following proposition answers this question.

Proposition 6.3.3. For granular approximations obtained with the p-quantile loss, the monotonicity property holds. More precisely, let p and q be two real numbers from the unit interval and let $\hat{A}_p$ and $\hat{A}_q$ be the outputs of the optimization problem (6.2) or (6.3) with p and q as probability parameters. It holds that

$$p \leq q \Rightarrow \forall u \in U, \hat{A}_p(u) \leq \hat{A}_q(u).$$

Proof. The proof is provided in Subsection 6.4.4. It relies on the greedy combinatorial approach presented in the previous subsections of Section 6.4, hence those previous sections are necessary to understand of the proof. □

In Proposition 6.3.3, we first notice that when $p = 0$, we have the rough lower approximation, and when $p = 1$, we have the rough upper approximation, according to Proposition 6.3.2. If $0 < p < 1$, we can obtain different approximations that lie between the lower and the upper one and which are ordered w.r.t. inclusion.

For the fuzzy rough approximations that are obtained with IMTL operators, we have the well known duality property as stated in Section 2.2.1. The following lemma and proposition extend that property to granularly representable sets and granular approximations. The duality property is particularly important for the binary classification problems.

It ensures that granular approximations of two different decision classes are complementary w.r.t. a given fuzzy negation N.

Lemma 6.3.1. If fuzzy set A is granularly representable w.r.t. T-preorder relation $\widetilde{R}$, then coA is granularly representable w.r.t. $\widetilde{R}^{-1}$.

Proof. For A being granularly representable, it holds that for all $u, v \in U$ we have

$$T(\widetilde{R}(u,v), A(v)) \leq A(u).$$

Applying negation N on both sides of the inequality, we have

$$\begin{aligned} T(\widetilde{R}(u,v), A(v)) \leq A(u) &\Rightarrow N(T(\widetilde{R}(u,v), A(v))) \geq N(A(u)) \\ &\Leftrightarrow I(\widetilde{R}(u,v), coA(v)) \geq coA(u) \\ &\Leftrightarrow T(coA(u), \widetilde{R}(u,v)) \leq coA(v) \\ &\Leftrightarrow T(\widetilde{R}^{-1}(v,u), coA(u)) \leq coA(v). \end{aligned}$$

The first equivalence follows from Proposition (2.6h) while the second is the residuation property. □

In the proof of Lemma 6.3.1, implication becomes an equivalence if we use IMTL triplets as operators.

Proposition 6.3.4. Let $\alpha_u, u \in U$ be a minimizer of the optimization problem (6.1) with nilpotent t-norm $T_{L,\varphi}$, relation $\widetilde{R}$, observations $\bar{A}$ and risk $\sum_{u \in U} L_p(\varphi(\bar{A}(u)), \alpha_u)$ (for short L_p problem). Then $1 - \alpha_u, u \in U$, is a minimizer of the optimization problem (6.1) with the same t-norm, relation $\widetilde{R}^{-1}$, observations $\bar{A}$ and risk $\sum_{u \in U} L_{1-p}(\varphi(co\bar{A}(u)), \alpha_u)$ (for short L_{1-p} problem).

Proof. Solution $\alpha_u, u \in U$, is a feasible solution of the L_p problem, i.e., it satisfies consistency conditions w.r.t. relation $\widetilde{R}$

$$\alpha_u - \alpha_v + 1 \geq \varphi(\widetilde{R}(u,v)).$$

The expression above is equivalent to

$$(1 - \alpha_v) - (1 - \alpha_u) + 1 \geq \varphi(\widetilde{R}^{-1}(v,u)),$$

which states that $1 - \alpha_u, u \in U$, satisfies the consistency conditions w.r.t. relation $\widetilde{R}^{-1}$ and, therefore, it is a feasible solution of the L_{1-p} problem.

We observe that $\varphi(co\bar{A}(u)) = \varphi(\varphi^{-1}(1-\varphi(\bar{A}(u)))) = 1-\varphi(\bar{A}(u))$. Regarding the empirical risk, we have that

$$\begin{aligned}
L_p(\varphi(\bar{A}(u)), \alpha_u) &= \begin{cases} p|\varphi(\bar{A}(u)) - \alpha_u| & \text{if } \varphi(\bar{A}(u)) - \alpha_u \geq 0, \\ (1-p)|\varphi(\bar{A}(u)) - \alpha_u| & \text{if } \alpha_u - \varphi(\bar{A}(u)) \geq 0, \end{cases} \\
&= \begin{cases} p|(1-\alpha_u) - (1-\varphi(\bar{A}(u)))| & \text{if } (1-\alpha_u) - (1-\varphi(\bar{A}(u))) \geq 0, \\ (1-p)|(1-\alpha_u) - (1-\varphi(\bar{A}(u)))| & \text{if } (1-\varphi(\bar{A}(u))) - (1-\alpha_u) \geq 0, \end{cases} \\
&= \begin{cases} (1-p)|\varphi(co\bar{A}(u))) - (1-\alpha_u)| & \text{if } \varphi(co\bar{A}(u)) - (1-\alpha_u) \geq 0, \\ p|\varphi(co\bar{A}(u)) - (1-\alpha_u)| & \text{if } (1-\alpha_u) - \varphi(co\bar{A}(u)) \geq 0, \end{cases} \\
&= L_{1-p}(\varphi(co\bar{A}(u)), 1-\alpha_u).
\end{aligned}$$

Due to previous equality, we have that non-optimal solution of the L_p problem different than $\alpha_u, u \in U$, will lead to the higher value of L_{1-p} loss. This means that $1-\alpha_u, u \in U$, as a feasible solution, is indeed an optimal solution. □

Since the optimal fuzzy set $\hat{A}$ of the L_p problem is calculated as $\hat{A}(u) = \varphi^{-1}(\alpha_u)$, then the optimal fuzzy set of the L_{1-p} is $\varphi^{-1}(1-\alpha_u) = \varphi^{-1}(1-\varphi(\hat{A}(u))) = N(\hat{A}(u)) = co\hat{A}(u)$, i.e., we have the duality.

The duality also holds for the mean squared error risk. The proof is very similar to the proof of Proposition 6.3.4 where the only difference is that the empirical risk stays the same in the dual problems.

We have the following corollary of Proposition 6.3.4

Corollary 6.3.1. Loss functions $L_{AEL,\varphi}$ and $L_{SEL,\varphi}$ are N-duality preserving for IMTL triplet $(T_{L,\varphi}, I_{L,\varphi}, N_{L,\varphi})$.

Proof. In the proof of Proposition 6.3.4, we showed that for some solution $\alpha_u, u \in U$ defined as $\alpha_u = \varphi(\hat{A}(u))$ and loss function L_{AEL}, it holds that

$$L_{AEL}(\varphi(\bar{A}(u)), \alpha_u) = L_{AEL}(\varphi(co\bar{A}(u)), 1-\alpha_u).$$

If we introduce notations $y = \bar{A}(u)$ and $\hat{y} = \hat{A}(u)$, we have that

$$L_{AEL}(\varphi(y), \varphi(\hat{y})) = L_{AEL}(\varphi(N_{L,\varphi}(y)), 1-\varphi(\hat{y})).$$

Furthermore, the previous expression is equivalent to:

$$\begin{aligned}
L_{AEL,\varphi}(y, \hat{y}) &= L_{AEL,\varphi}(N_{L,\varphi}(y), \varphi^{-1}(1-\varphi(\hat{y}))) \\
\Leftrightarrow L_{AEL,\varphi}(y, \hat{y}) &= L_{AEL,\varphi}(N_{L,\varphi}(y), N_{L,\varphi}(\hat{y})),
\end{aligned}$$

which is the N-duality preserving property. Using the analogy, we can prove the same for $L_{SEL,\varphi}$. □

6.4 Granular approximations and the minimum-cost flow problem

In this section, we discuss the combinatorial approach to the optimization problems (6.2) and (6.3). We first formulate their dual versions, and model the dual of (6.2) as the min-cost flow problem while the dual of (6.3) takes a form of the slightly modified min-cost flow problem. We provide combinatorial algorithms to solve the newly formulated problems, prove the correctness of those algorithms and provide the proofs of certain propositions from 6.3 using the new formulations.

6.4.1 Introduction to the minimum-cost flow problem

This subsection is based on the monograph [1], especially on its 9th chapter.

A flow network is defined as a directed graph where a real value called imbalance is assigned to each node. Imbalances split nodes into two subsets: supply nodes with a positive imbalance (supply value) and demand nodes with a negative imbalance (demand value). Moreover, each edge is characterized by a positive real capacity, and a cost value. We also assign flow amounts to each edge which satisfy the condition that they are at most as large as capacities. More formally, let G be a finite set of nodes, $E \subseteq G \times G$ the finite set of edges, while $F = (G, E)$ is the flow network. We denote imbalances with b_i for $i \in G$, capacities with $l_{i,j}$, costs with $c_{i,j}$ and flow with $z_{i,j}$ for $(i, j) \in E$.

The minimum-cost flow problem is an optimization problem defined on a flow network where we want to transport flow from the supply nodes to the demand nodes, such that

- the difference between the flow that leaves a node and the flow that enters the node is equal to the imbalance of this node,
- a flow in a particular edge is at most as large as the capacity of that edge, and
- the total cost of the flow transportation is minimal.

Formally, we have the following problem:

$$\text{minimize} \quad \sum_{(i,j)\in E} c_{i,j} z_{i,j}, \tag{6.6a}$$

$$\text{subject to} \quad \sum_{j:(i,j)\in E} z_{i,j} - \sum_{j:(j,i)\in E} z_{j,i} = b_i, \qquad i \in G \tag{6.6b}$$

$$0 \leq z_{i,j} \leq l_{i,j}. \qquad (i,j) \in E \tag{6.6c}$$

We distinguish two sets of constraints in the previous optimization problem: balance constraints (6.6b) and capacity constraints (6.6c). If we sum the balance constraints, we get $\sum_{i\in G} b_i = 0$ which states that the amount of supply is equal to the amount of demand, which is a necessary assumption to have a feasible solution.

We say that a flow is feasible if it is a feasible solution of (6.6), while we say that we have a pseudo-flow if only the capacity constraints are satisfied.

For a given pseudo-flow z', a residual network $F' = (G, E')$ can be defined. We have new imbalances:

$$b'_i = b_i - \left(\sum_{j:(i,j)\in E} z_{i,j} - \sum_{j:(j,i)\in E} z_{j,i} \right),$$

while for each edge $(i,j) \in E$ for which $z'_{i,j} > 0$, we add the reverse edge (j,i) to the network with cost $c'_{j,i} = -c_{i,j}$, while keeping the original edge. The capacity of the original edge (i,j) in F' is $l'_{i,j} = l_{i,j} - z_{i,j}$, while the capacity of the added reverse edge (j,i) is $l'_{j,i} = z_{i,j}$. We may notice that when adding a new edge (j,i) to E', there can already exist an edge (j,i) from E. However, in our case of use, we will not face such an issue, i.e., we will have either (i,j) or (j,i) in E and not both at the same time. The residual network keeps the complete information about flow z' which can be reconstructed from F'.

The concept of residual network is important for the development of algorithms for solving (6.6). In this moment, we will not discuss the existence of a feasible solution in general since later we will show that it always exists in our case of use.

A cost of a particular path or cycle in the flow network is calculated as the sum of the costs of edges in that path or cycle. For an optimal flow z^*, we have the following result.

Proposition 6.4.1. A flow z^* is optimal if and only if there are no cycles of negative cost in the residual network $F(z^*)$.

Bearing in mind Proposition 6.4.1, a simple algorithm can be constructed to solve (6.6). Namely, we construct an initial feasible flow in our network, then search for the negative cycles and eliminate them.

However, a more useful algorithm for us is the Successive Shortest Path (SSP) algorithm for solving the minimum-cost flow problem. The algorithm is provided as Algorithm 1.

Algorithm 1 Successive Shortest Paths

1: **Input:** Flow network $F = (G, E)$.
2: **Output:** Flow z.
3: Set initial flow $z_{i,j} = 0$, $(i, j) \in E$
4: Set initial residual network $F' = F$
5: **while** there exist supply/demand values different from 0 **do**
6: Pick supply node i and demand node j
7: Calculate the shortest path P from i to j using cost values from F'
8: Send the largest possible amount of flow through P
9: Update F'
10: Reconstruct z from F'

The shortest path P can be calculated using the Bellman-Ford algorithm since F' may contain negative values. The largest possible amount of flow through P is calculated as

$$\delta = \min\{b'_i, |b'_j|, c'_{i_1,j_1} \text{ for } (i_1, j_1) \in P\}.$$

The residual network is then updated such that

- $b'_i = b'_i - \delta$, $b'_j = b'_j + \delta$
- $c'_{i,j} = c'_{i,j} - \delta$, $c'_{j,i} = c'_{j,i} + \delta$ for $(i, j) \in P$

The idea of the proof of correctness is that sending a flow through the shortest path does not produce negative cycles in the residual network. Hence, when all supply is sent to the demand nodes and the feasible solution is achieved, it will be an optimal one.

We also introduce generalized network flows based on Chapter 15 of [1]. In some cases, the flow in a particular edge may be increased or decreased by a multiplier after it leaves the left node of the edge. Denote the multipliers with $m_{i,j}$ for $(i, j) \in E$. The generalized minimum-cost

flow problem is then formulated as

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,j)\in E} c_{i,j} z_{i,j}, \\ \text{subject to} \quad & \sum_{j:(i,j)\in E} m_{i,j} z_{i,j} - \sum_{j:(j,i)\in E} z_{j,i} = b_i, \quad i \in G \\ & 0 \le z_{i,j} \le l_{i,j}, \qquad\qquad (i,j) \in E. \end{aligned} \tag{6.7}$$

If the multiplier is greater than 1, then the flow is increased while if it is smaller than 1, then the flow is decreased.

Different theoretical results hold for the generalized minimum-cost flow problem (6.7). Fortunately, our particular case of (6.7) allows obtaining similar properties as in the ordinary minimum-cost flow problem (6.6).

6.4.2 Duality and the combinatorial approach

In this subsection, the dual optimization problems of (6.2) and (6.3) are considered. In our particular case, the dual problems are interesting since they can be modeled using graph theory and can be solved using combinatorial optimization methods. These combinatorial algorithms may not be more efficient than the simplex method used for solving linear programs, but their development is important since they allow us to prove some interesting properties of the estimated fuzzy set. We examine optimization problem (6.2). First, we eliminate variables $x_u, u \in U$, using constraints $x_u = y_u + \bar{A}_\varphi(u) - \alpha_u$ and we denote $M(u,v) = 1 - \widetilde{R}_\varphi(u,v)$. Then, the problem is reformulated as

$$\begin{aligned} \text{maximize} \quad & p \sum_{u\in U} \alpha_u - \sum_{u\in U} y_u, \\ \text{subject to} \quad & \alpha_v - \alpha_u \le M(u,v), \quad u,v \in U \\ & \alpha_u - y_u \le \bar{A}_\varphi(u), \quad u \in U \\ & y_u \ge 0 \qquad\qquad u \in U. \end{aligned} \tag{6.8}$$

Its dual problem is then

$$\begin{aligned} \text{minimize} \quad & \sum_{u,v\in U} M(u,v) z_{u,v} + \sum_{u\in U} \bar{A}_\varphi(u) z_{0,u} \\ \text{subject to} \quad & -z_{0,u} + \sum_{v\in U} z_{u,v} - \sum_{v\in U} z_{v,u} = -p, \quad u \in U \\ & z_{0,u} \le 1. \qquad\qquad u \in U. \end{aligned} \tag{6.9}$$

In (6.9), variables $z_{u,v}, u, v \in U$, correspond to the first set of constraints from primal (6.8), while variables $z_{0,u}, u \in U$, correspond to the second set of constraints from the primal. The first set of constraints in (6.9) corresponds to variables $\alpha_u, u \in U$, from the primal, while the second set of constraints corresponds to variables $y_u, u \in U$, from the primal.

If we sum up the equality constraints, we get $\sum_{u \in U} z_{0,u} = np$ where $n = |U|$. Bearing this in mind, we see that (6.9) is exactly the minimum-cost flow problem on $n+1$ nodes where we have one supply node with imbalance $b_0 = np$ and n demand nodes with imbalances $-p$. From the supply node, to all other nodes we have flow $z_{0,u}$, costs $\bar{A}_\varphi(u)$, while all capacities are equal to 1. Among the demand nodes, there is a flow $z_{u,v}, u, v \in U$, costs $M(u,v)$, and there are no capacity constraints.

To make our model even simpler, we utilize the T-transitivity of the relation $\widetilde{R}$. It is easy to verify that the T-transitivity is equivalent to $M(u,v) + M(v,w) \geq M(u,w)$ for $u, v, w \in U$. Using this fact, we have that there is an optimal flow which does not use two consecutive edges that are between demand nodes. Assume that for an optimal flow z^* we have $z^*_{u,v} > 0$ and $z^*_{v,w} > 0$, and let $\delta = \min(z^*_{u,v}, z^*_{v,w})$. Then the flow $z^*_{u,v} - \delta, z^*_{v,w} - \delta, z^*_{u,w} + \delta$ is feasible and at most as expensive as the previous flow, i.e., it is optimal. The new flow does not use two consecutive edges since either $z^*_{u,v} - \delta$ or $z^*_{v,w} - \delta$ is 0. The previous elaboration further implies that an optimal flow from the supply node can travel through at most one intermediary node to the destination demand node. Hence, our initial network flow on $n+1$ nodes can be transformed into a flow network on $2n+1$ nodes which has the form of a bipartite graph plus the supply node. One independent set in the bipartite graph is formed by the intermediate nodes, while the other independent set is formed by the destination nodes.

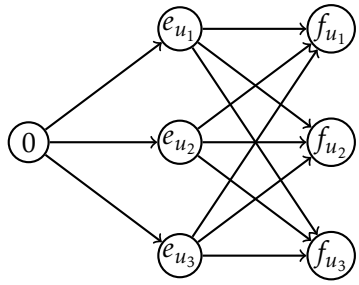

Figure 6.1: Flow modeled as a bipartite graph

In Figure 6.1, we have an example of a bipartite network on set of instances $U = \{u_1, u_2, u_3\}$. Since $n = 3$ in this case, the bipartite graph has $2 \cdot 3 + 1 = 7$ nodes. Node 0 is the supply node with imbalance np. Nodes $\{e_{u_1}, e_{u_2}, e_{u_3}\}$ are the intermediate nodes without imbalances while $\{f_{u_1}, f_{u_2}, f_{u_3}\}$ are the destination nodes with demands $-p$. For $u \in U$, the cost of edges $(0, e_u)$ is $\bar{A}_\varphi(u)$ while the capacity is 1. For $u, v \in U$, the cost of edges (e_u, f_v) is $M(u, v)$ while the capacity is unbounded. The cost of edges (e_u, f_u) is then 0. If a flow takes path $(0, e_u, f_v)$ in the bipartite graph for $u, v \in U$, and $u \neq v$, then in the original network it means that the flow travels from 0 to v using intermediate node u. If $u = v$, it means that there were no intermediate nodes and that the flow travels directly from 0 to u.

For a given flow in a bipartite network flow, there is also the corresponding residual network. In such a residual network, there are edges from the destination nodes to the intermediate nodes and from the intermediate nodes to the supply. The costs and the capacities of the new edges are then calculated as was explained in Section 6.4.1.

The bipartite network representation is useful from the perspective of the flow decomposition. For a feasible flow, it is easy to represent it as a sum of simple flows that go from the supply node to the destination node. In the original network, one node can be a destination node for some flow but also an intermediate node for a different flow. Hence, the decomposition is harder in the original network. The decomposition will be important later when dealing with the dual of (6.3).

The next question is how to reconstruct optimal solution of the primal problem, i.e., to calculate α^* from a solution of the dual z^*. Following the duality theory provided in [1], an optimal vector α^* can be obtained as lengths of shortest paths from the supply node to the corresponding destination nodes in the residual network of z^*.

Now, we examine the dual of (6.3). The linear program here can be rewritten similarly as (6.8), just with the different granularity constraints. Here instead of $\alpha_v - \alpha_u \leq M(u, v)$ we have $\alpha_v \widetilde{R}_\varphi(u, v) \leq \alpha_u$. The dual of such formulated problem is then

$$\begin{aligned} \text{minimize} \quad & \sum_{u \in U} \bar{A}_\varphi(u) z_{0,u}, \\ \text{subject to} \quad & -z_{0,u} + \sum_{v \in U} z_{u,v} - \sum_{v \in U} \widetilde{R}_\varphi(v, u) z_{v,u} = -p, \quad u \in U \\ & z_{0,u} \leq 1. \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad u \in U \end{aligned} \tag{6.10}$$

The difference between (6.9) and (6.10) is that in the latter, we have multipliers $\widetilde{R}_\varphi(u,v), u,v \in U$, instead of costs on the edges. More precisely a flow that goes from node v to node u will be multiplied with $\widetilde{R}_\varphi(u,v)$. For that purpose, we introduce the new notation for multipliers $J(u,v) = \widetilde{R}_\varphi(u,v)$, in order to distinguish the contexts of fuzzy relations and flow networks and to be able to denote the multipliers on paths, not only on edges. Due to the multipliers, we now deal with the minimum-cost flow problem on a generalized flow network with $n+1$ nodes among which there are n demand nodes with demand $-p$ and one supply node with an unspecified amount of supply.

We may notice that in this case the edges of the network consist of two different groups. The first group is formed by the edges from the supply nodes to the demand nodes. These edges have costs and do not have multipliers. The second group is formed by the edges among the demand nodes. These edges, conversely, have multipliers and do not have costs. Similarly to (6.9), we are able to utilize the T-transitivity of $\widetilde{R}_\varphi$ w.r.t. T_P in a way that there is an optimal flow which does not use two consecutive edges from the second group. If we have three demand nodes $u,v,w \in U$ in a network and an optimal flow that uses edges (u,v) and (v,w), we can redirect the flow to use only edge (u,w) and the redirected flow will have smaller or equal loss than the original flow. This will further lead to a smaller or equal cost of the redirected flow which makes it optimal. Therefore, as above, there is an optimal solution in which a flow travels from the supply node to the destination demand node using at most one intermediate node. This again further implies that the initial general network on $n+1$ nodes can be transformed into a generalized bipartite flow network on $2n+1$ nodes. For the new network, the same model applies as in Figure 6.1. Using this model, we can clearly see the difference between the two groups of edges introduced above. The first group is formed by the edges between the supply node and the left partition of the bipartite graph (intermediate nodes), while the second group is formed by the edges between the two partitions of the bipartite graph.

As before, for a given flow on the generalized bipartite network, we have the corresponding residual network. The same properties apply as above except in the case when the flow passes through an edge with multiplier. In that case, if the original edge has multiplier $J(u,v)$ then the reverse edge in the residual network will have multiplier $\frac{1}{J(u,v)}$ which is an edge of a gain type (greater than 1).

We will now construct a new algorithm for solving a generalized

minimum-cost flow problem on a generalized bipartite flow network. The algorithm is based on the existing SSP algorithm presented in Algorithm 1. Assume that we have a demand node f_u to which we want to deliver some flow b. We want to deliver the flow at the cheapest possible price. If we deliver a flow using intermediate node e_v, then the amount of flow that we have to take from the supply node is $\frac{b}{J(v,u)}$ and the cost of such flow is $\frac{b\bar{A}_\varphi(v)}{J(v,u)}$. In general, a price to deliver a unit of flow is a ratio of the cost of an edge from the supply to the first partition and the product of multipliers of edges that connect the two partitions. Bear in mind that in the residual network, a flow may use multiple edges between partitions (edges with multipliers) to deliver the flow. Using this, we construct the greedy approach presented as Algorithm 2.

Algorithm 2 Generalized successive shortest paths

1: **Input:** Bipartite flow network F.
2: **Output:** Flow z.
3: Set initial flow $z_{i,j} = 0$, $(i, j) \in E$
4: Set initial residual network $F' = F$
5: **while** there exists demand value different than 0 **do**
6: Pick a demand node i
7: Calculate the smallest possible cost from the supply node to i
8: Calculate the largest amount of flow that can be sent through the least costly path
9: Send the calculated flow through the least costly path
10: Update F'
11: Reconstruct z' from F'

To calculate the smallest possible cost from the supply node, we can use a shortest path method. We want to minimize the ratio of one cost value (from the supply to the intermediate nodes) and a product of multipliers (between intermediate and destination nodes). If we apply logarithms on the cost values and reciprocals of the multipliers, we may apply the Bellman-Ford algorithm to calculate the shortest path between the supply node and the chosen demand node in order to obtain a least costly way to transport the flow.

After the shortest path is determined, we have to calculate the amount of flow that will be taken from the supply node in order to deliver the maximal amount of flow to the demand node. In comparison with the standard minimum-cost flow problem, here we have

to take into account all the losses and gains that happen during the flow transfer. Denote the shortest path in the residual network with $P = (0, e_{u_1}, f_{u_2}, e_{u_3}, \ldots, f_{u_k})$ and let b be a demand of node f_{u_k}. We would like to deliver $|b|$ ($|\cdot|$ stands for absolute value) amount of flow to the demand node from the supply node, but this is not always possible due to the capacities of particular edges on path P. The maximal amount of flow can be determined recursively. The maximal amount of flow that can be transferred from node $f_{u_{k-2}}$ to node f_{u_k} is bounded by the capacity of the reverse edge $l'_{f_{u_{k-2}}, e_{u_{k-1}}}$ and the demand divided with the loses on the edges in between $\frac{|b|J(u_{k-1}, u_{k-2})}{J(u_{k-1}, u_k)}$. Using that reasoning, if we set the initial value $z' = |b|$, then we can use the following iteration formula.

$$z' = \min\left(\frac{z'J(u_{k-2i+1}, u_{k-2i})}{J(u_{k-2i+1}, u_{k-2i+2})}, l'_{f_{u_{k-2i}}, e_{u_{k-2i+1}}}\right),$$

for i going from 1 to $\frac{k}{2} - 1$. The last step is $z' = \min(\frac{z'}{J(u_1, u_2)}, l'_{0, e_{u_1}})$ for subpath $(0, e_{u_1}, f_{u_2})$.

After z' is calculated, we have to determine the amount of flow that will end up in the demand node f_{u_k} as well as to update the residual network on path P. In the first step, z' leaves the supply node, passes node e_{u_1} and enters node f_{u_2}. On edge (e_{u_1}, f_{u_2}) it was multiplied with $J(u_1, u_2)$: $z' = J(u_1, u_2)z'$. Then we update the residual network on edges (f_{u_2}, e_{u_1}) and (f_{u_2}, e_{u_3}): $l'_{f_{u_2}, e_{u_1}} = l'_{f_{u_2}, e_{u_1}} + z'$, $l'_{f_{u_2}, e_{u_3}} = l'_{f_{u_2}, e_{u_3}} - z'$ and we send the flow to the next node from the second partition and repeat the process. After the remaining flow arrives to the demand node, we increase the imbalance of the demand node.

Since Algorithm 2 is novel, we cannot benefit from the existing theory as we did in the case of Algorithm 1. In Section 6.4.3, we will show that Algorithm 2 indeed returns an optimal result, as well as how to construct a solution of the primal problem from the solution of the dual one. As is shown in Section 6.4.3, α^* is constructed by performing step 7 (without logarithms) of Algorithm 2 on the residual network of z^*, i.e., it is the smallest possible cost of the transport from the supply node to the destination nodes.

6.4.3 Proof of correctness for Algorithm 2

In this subsection we prove that Algorithm 2 terminates and that it outputs an optimal solution. Also, we construct a way to obtain a solution of the primal problem from the solution of the dual one.

We first prove the termination.

Proposition 6.4.2. Assume that all parameters in Algorithm 2 are rational numbers. Then algorithm 2 terminates.

Proof. It is easy to see that if we multiply the right side of the constraints in (6.10) with a positive constant C, the optimal solution is Cz^* where z^* is the solution of the initial problem. For some parameter a in (6.10) we have its rational representation $a = \frac{q}{r}$ for q and r being integers. Let C be the least common multiple (LCM) of all integers q and r for all parameters in (6.10). If we multiply the right side of the constraints in (6.10) with C, then all the demand values will become integers and all intermediate flows in Algorithm 2 will become integers. That further implies that all the updates on demands in Algorithm 2 will be integers which further implies that the algorithm will terminate in at most Cpn steps. □

In practice, the termination is always guaranteed since computers can work only with rational numbers.

Now, let us define a flow cycle in the residual generalized bipartite network. The cycle starts with an edge from the first part (costly edges without multipliers) of the network, then it contains edges from the second part (edges with multipliers without costs) and ends with a reverse edge from the first part. A model of such cycle is shown in Figure 6.2.

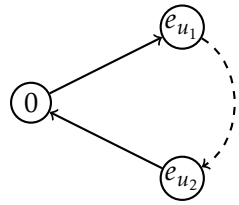

Figure 6.2: Cycle in a generalized bipartite network

In Figure 6.2, the dashed line between e_{u_1} and e_{u_2} stands for the subpath that contains only the edges from the second part of the residual network. Also, it may hold that $e_{u_1} \equiv e_{u_2}$. In that case, the cycle consists only of the edges from the second part. Let $J(e_{u_1}, \ldots, e_{u_2})$ be a multiplier of the path that consists of the edges from the second part of the residual network, i.e., a product of the multipliers on the edges from the path. We say that the cycle is of negative cost if $A_\varphi(u_1) < J(e_{u_1}, \ldots, e_{u_2}) A_\varphi(u_2)$. As a reminder, $A_\varphi(u_1)$ and $A_\varphi(u_2)$ are the costs on edges $(0, e_{u_1})$ and $(0, e_{u_2})$. The reason why the cycle is of negative cost is that if we send a unit of

flow along it, the cost of that flow is $A_\varphi(u_1) - J(e_{u_1}, \ldots, e_{u_2})A_\varphi(u_2)$, i.e., the cost is negative. Such flow would not change any demand value on the destination nodes but it will reduce the overall cost of the flow.

The next proposition utilizes the bipartite representation of the flow network.

Proposition 6.4.3. Every flow in a generalized bipartite network can be represented as a sum of a finite number of simple path flows from the supply node to a destination node.

Proof. Let z be a flow and consider an edge (e_{u_1}, f_{u_2}) from the second part of the network and its flow value $z_{e_{u_1}, f_{u_2}}$. This edge receives a flow from edge $(0, e_{u_1})$ which is a part of path flow z_P from path $P = (0, e_{u_1}, f_{u_2})$ that connects the supply node and the destination node f_{u_2}. z_P is then a summand in the representation while the remaining flow $z - z_P$ has no flow on the edge (e_{u_1}, f_{u_2}) and hence we can remove that edge from the network flow. If we continue, in every step we will construct one summand and remove one edge from the second part of the network. Since we have a finite number of edges, we have a finite number of summands. □

We have the following result.

Proposition 6.4.4. Solution z^* is optimal in the generalized bipartite network if and only if its residual network does not contain negative cost cycles.

Proof. ($\Rightarrow$) When the solution is optimal, there are no negative cost cycles. If otherwise, we could send a flow through a negative cost cycle and we would decrease the cost of the overall flow as described above. That contradicts the optimality.

($\Leftarrow$) Assume that z^* is a feasible solution whose residual network does not contain negative cost cycles and let z' be a feasible solution. Let $z' = z^* + z''$. We first show that z'' is a feasible flow from the residual network of z^*, i.e., it satisfies its constraints. For an edge $(0, e_{u_1})$ if the flows are different, we can have either $z'_{0,e_{u_1}} > z^*_{0,e_{u_1}}$ or $z'_{0,e_{u_1}} < z^*_{0,e_{u_1}}$. In the first case, it holds that $z'_{0,e_{u_1}} = z^*_{0,e_{u_1}} + z''_{0,e_{u_1}}$, i.e., $z''_{0,e_{u_1}}$ uses the original edge. Since $z'_{0,e_{u_1}} \leq 1$ then $z''_{0,e_{u_1}} \leq 1 - z^*_{0,e_{u_1}}$ which is a constraint from the residual network. In the second case, it holds that $z'_{0,e_{u_1}} = z^*_{0,e_{u_1}} - z''_{e_{u_1},0}$, i.e., $z''_{e_{u_1},0}$ uses the reverse edge. Since $z'_{0,e_{u_1}} \geq 0$ then $z''_{e_{u_1},0} \leq z^*_{0,e_{u_1}}$ which is a constraint for the reverse edge from the residual network. Using similar reasoning, we can conclude the same for the whole network.

The next step is to show that z'' is a sum of a finite number of simple flow cycles, as shown in Figure 6.2, i.e., it has a cycle representation. Proposition 6.4.3 states that both flows z' and z^* are sums of simple flows on paths from the supply node to a destination node. Take a summand z'_{P_1} of z' and summand $z^*_{P_2}$ of z^* for $P_1 = (0, e_{u_1}, f_{u_3})$ and $P_2 = (0, e_{u_2}, f_{u_3})$. The paths have the same destination node. Assume that the first summand delivers amount b_1 of flow to the destination node while the second delivers amount b_2 of flow to the same node. W.L.O.G. assume that $b_1 \geq b_2$. Then the flow $\frac{b_2}{b_1} z'_{P_1} - z^*_{P_2}$ is a flow along cycle $(0, e_{u_1}, f_{u_3}, e_{u_2}, 0)$ and one of the summands in the cycle representation of z''. After the summand is identified, we remove its flow from the consideration. In that moment, $z^*_{P_2}$ is fully removed while we are left with $(1 - \frac{b_2}{b_1}) z'_{P_1}$ from the first path. We continue to create flow cycles as summands from the remaining path flows from z' and z^*. Since after every summand is identified we remove one path flow, the number of summands is finite. Hence, z'' is a sum of a finite number of cycle flows. Since z'' is a flow in the residual network of z^*, all the cycles from its cycle representation are of positive cost by the assumption which implies that z'' is of positive cost. Since the cost of z' is a sum of costs of z^* and z'', cost of z' is larger than the cost of z^*. Since flow z' was an arbitrary feasible flow, we conclude that z^* is an optimal flow. □

Proposition 6.4.5. Algorithm 2 returns an optimal solution.

Proof. Assume that in one iteration of Algorithm (2), the shortest path had the form $P_1 = (0, e_{u_2}, \ldots, f_{u_3})$ and that after the step, the negative cost cycle $(0, e_{u_1}, \ldots, f_{u_3}, \ldots, e_{u_2}, 0)$ was formed. The negative cost cycle is formed from the path $P_2 = (0, e_{u_1}, \ldots, f_{u_3})$ and the reverse path P_1. The model of such cycle is represented in Figure 6.3.

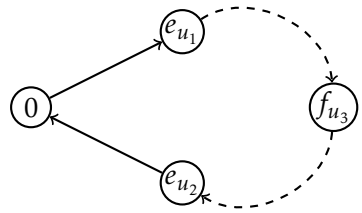

Figure 6.3: Cycle after one step of Algorithm 2

The dots in the cycle as well as dashed edges in the figure stand for edges from the second part of the residual network (edges with

multipliers). If the cycle is negative, then it holds that $A_\varphi(u_1) < J(e_{u_1},\dots,f_{u_3})J(f_{u_3},\dots,e_{u_2})A_\varphi(u_2)$. The latter is equivalent to $\frac{A_\varphi(u_1)}{J(e_{u_1},\dots,f_{u_3})} < \frac{A_\varphi(u_2)}{J(e_{u_2},\dots,f_{u_3})}$ which states that path P_2 is actually shorter than P_1 which contradicts the assumption that P_1 is the shortest path at this step.

Hence, at every iteration of Algorithm 2, there are no negative cost cycles and as soon as the feasible solution is achieved, it will be an optimal one according to Proposition 6.4.4. □

After we constructed the algorithm that solves the dual optimization problem, we need to obtain an optimal solution for the primal which was our initial goal. First, we need one technical proposition.

Proposition 6.4.6. For a given generalized bipartite network, there exists an optimal solution z^* for which it holds

$$z^*_{0,e_u} > 0 \implies z^*_{e_u,f_u} > 0.$$

Proof. Assume that for some solution z^* and some instance u we have that $z^*_{0,e_u} > 0$ and $z^*_{e_u,f_u} = 0$. Then, in the simple path decomposition of the flow, we have path $(0, e_v, f_u)$ that delivers flow to f_u, and path $(0, e_u, f_w)$ that uses flow from edge $(0, e_u)$. Then, in the residual network of z^*, $C = (e_u, f_u, e_v, f_w, e_u)$ is a cycle. Due to transitivity of $\widetilde{R}$, it holds that

$$J(v,u)J(u,w) \le J(v,w).$$

If $J(v,u)J(u,w) < J(v,w)$, then C is a negative cost cycle which contradicts the optimality of z^*. If $J(v,u)J(u,w) = J(v,w)$ then cycle C is a zero-cost cycle and a flow can be sent through the cycle without violating optimality. Hence, sending some amount of flow through the cycle, we will construct another optimal solution z^{**} where $z^{**}_{e_u,f_u} > 0$. □

In practice, if we obtain an optimal solution containing an edge for which the previous proposition does not hold, we can get another optimal solution, without such edges, as explained in the proof of the previous proposition. From now on, we assume that we have an optimal solution for which the previous proposition holds.

We continue with the duality theory of the linear programs.

According to the strong duality theorem [96], if there exists an optimal solution of the dual problem z^* then, there exists an optimal solution

for the primal problem α^*, and it holds that the values of objectives in (6.8) and in (6.10) are equal, i.e.,

$$\sum_{u\in U} \bar{A}_\varphi(u) z^*_{0,u} = \sum_{u\in U} p\alpha^*_u - \sum_{u\in U} \max(\alpha^*_u - \bar{A}_\varphi(u), 0). \tag{6.11}$$

In the previous expression, y_u is replaced with its definition. In an optimal solution, for $u \in U$, we have that

$$\sum_{v\in U} z^*_{u,v} = z^*_{0,u}, \quad \sum_{v\in U} \widetilde{R}_\varphi(v,u) z^*_{v,u} = p. \tag{6.12}$$

We have the following equalities:

$$\begin{aligned}
\sum_{u\in U} \max(\alpha^*_u - \bar{A}_\varphi(u), 0) &= \sum_{u\in U} p\alpha^*_u - \sum_{u\in U} \bar{A}_\varphi(u) z^*_{0,u} \\
&= \sum_{u\in U} p\alpha^*_u - \sum_{u\in U} (\bar{A}_\varphi(u) - \alpha^*_u) z^*_{0,u} - \sum_{u\in U} \alpha^*_u z^*_{0,u} \\
&= \sum_{u\in U} p\alpha^*_u - \sum_{u\in U} (\bar{A}_\varphi(u) - \alpha^*_u) z^*_{0,u} - \sum_{u\in U} \alpha^*_u \sum_{v\in U} z^*_{u,v} \\
&= \sum_{u\in U} p\alpha^*_u - \sum_{u\in U} (\bar{A}_\varphi(u) - \alpha^*_u) z^*_{0,u} \\
&\quad - \sum_{u,v\in U} (\alpha^*_u - \widetilde{R}_\varphi(u,v)\alpha^*_v) z^*_{u,v} - \sum_{u,v\in U} \widetilde{R}_\varphi(u,v)\alpha^*_v z^*_{u,v} \\
&= \sum_{u\in U} p\alpha^*_u - \sum_{u\in U} (\bar{A}_\varphi(u) - \alpha^*_u) z^*_{0,u} \\
&\quad - \sum_{u,v\in U} (\alpha^*_u - \widetilde{R}_\varphi(u,v)\alpha^*_v) z^*_{u,v} - \sum_{v\in U} \alpha^*_v \sum_{u\in U} \widetilde{R}_\varphi(u,v) z^*_{u,v} \\
&= \sum_{u\in U} (\alpha^*_u - \bar{A}_\varphi(u)) z^*_{0,u} - \sum_{u,v\in U} (\alpha^*_u - \widetilde{R}_\varphi(u,v)\alpha^*_v) z^*_{u,v}.
\end{aligned}$$

The second equality holds because of the left expression in (6.12) while the last equality holds because the right expression in (6.12). We have that for all $u \in U$, $\max(\alpha^*_u - \bar{A}_\varphi(u), 0) \geq (\alpha^*_u - \bar{A}_\varphi(u)) z^*_{0,u}$ and that for all $u, v \in U$, $\alpha^*_u - \widetilde{R}_\varphi(u,v)\alpha^*_v \geq 0$, since α^* is a feasible solution. Hence, for the previous equality to hold, we need to have that for all $u \in U$, $\max(\alpha^*_u - \bar{A}_\varphi(u), 0) = (\alpha^*_u - \bar{A}_\varphi(u)) z^*_{0,u}$ and that for all $u, v \in U$, $(\alpha^*_u - \widetilde{R}_\varphi(u,v)\alpha^*_v) z^*_{u,v} = 0$. The latter is equivalent to the following set of conditions.

- $z^*_{0,u} = 0 \implies \alpha^*_u \leq \bar{A}_\varphi(u)$,

- $0 < z^*_{0,u} < 1 \implies \alpha^*_u = \bar{A}_\varphi(u)$,
- $z^*_{0,u} = 1 \implies \alpha^*_u \geq \bar{A}_\varphi(u)$,
- $z^*_{u,v} > 0 \implies \alpha^*_u - \widetilde{R}_\varphi(u,v)\alpha^*_v = 0$,

for $u, v \in U$. We have the following conclusion: if we solve the dual optimization problem and obtain an optimal solution z^*, then a solution of the primal optimization problem is any α^* which satisfies the conditions listed above.

Moreover, α^* can be constructed by performing step 7 of Algorithm 2 on the residual network of z^*, i.e., it is the smallest possible cost of the transport from the supply node to the destination nodes. It is easily verifiable that such α^* satisfies the conditions above. The proof of this verification lies in that if we assume that some condition is not satisfied, then we would have a negative cost cycle which contradicts the optimality of z^*. To prove the contradiction, we need Proposition 6.4.6.

6.4.4 Proof of Proposition 6.3.3

Let $\alpha^p_u = \varphi(\hat{A}_p(u))$ and $\alpha^q_u = \varphi(\hat{A}_q(u))$ for $u \in U$. Then $\hat{A}_p(u) \leq \hat{A}_q(u) \Leftrightarrow \alpha^p_u \leq \alpha^q_u$. To prove this proposition, we will use Algorithm 1 in case of T_L and Algorithm 2 in case if T_P. We apply both algorithms on the bipartite flow network in the way that we first deliver amount p of flow to every destination node, then we calculate α^p as the smallest cost from the supply node to the destination nodes in the residual network, then we deliver additional amount $q - p$ of flow to every destination node and then we calculate α^q in the same way as α^p. Using this procedure, we may notice that to calculate α^q we need a few more iterations of the algorithms after α^p. Bearing this in mind, it is enough to prove that after every iteration of the algorithm, i.e., after sending some amount of flow to a destination node and updating the residual network, the cost from the supply node to every destination node stayed the same or is increased.

When updating residual network F', the possible changes in the residual networks are the following:

- Reverse edges between the supply node and the intermediate nodes can be added while the original edges can be removed.
- Reverse edges between the intermediate and destination nodes can be added or removed.

Adding reverse edges between the supply node and intermediate nodes is not important in this case, since shortest paths do not use these edges. Removing the original edges between the same nodes will not reduce the costs since the shortest paths now chose among the smaller set of edges. The same holds if we remove reverse edges between the intermediate nodes.

The last step is to prove that adding reverse edges between the intermediate and destination nodes will not reduce the costs from the supply to the destination nodes.

For that purpose, we consider Figure 6.4.

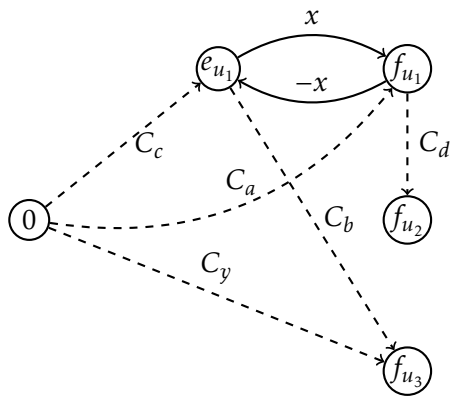

Figure 6.4: Flow modeled as a bipartite graph

With dashed lines, we denote certain paths for which the costs are marked on the figure. In both cases of T_L and T_P, the costs are the values used to calculate the shortest paths. Assume that in step i, we were calculating the shortest path between 0 and f_{u_2} and we obtained that the shortest path is $(0,\ldots,e_{u_1},f_{u_1},\ldots,f_{u_2})$ and since some flow is sent through that path, a reverse edge (f_{u_1},e_{u_1}) is created with cost $-x$. Assume that before step i, the shortest path from 0 to f_{u_3} was $(0,\ldots,f_{u_3})$ with cost C_y while after the previous step and after adding reverse edge (f_{u_1},e_{u_1}) the shortest path is $(0,\ldots,f_{u_1},e_{u_1},\ldots,f_{u_3})$ with cost $C_a - x + C_b$. Then, we have that $C_a + C_b < x + C_y$. Since the shortest path in step i was $(0,\ldots,e_{u_1},f_{u_1},\ldots,f_{u_2})$, it holds that $C_c + x \leq C_a$. Adding this to the previous expression, we have that

$$x + C_y > C_a + C_b \geq C_c + x + C_b \Leftrightarrow C_y > C_c + C_b.$$

The last inequality contradicts the assumption that before step i, the smallest cost between 0 and f_{u_3} is C_y.

6.5 Conclusion

In this chapter, we introduced a novel statistical learning approach for handling inconsistencies in prediction problems with respect to a fuzzy relation. Our work was motivated by the method introduced by Kotłowski and Słowiński [89] for handling monotone inconsistencies and we showed that the novel approach is a generalization of the same method in the binary classification case. Using fuzzy relations, the novel method is able to handle gradual relationships among instances while the KS approach can distinguish only two cases: either instances relate or not.

Our approach produces a granular approximation of a fuzzy set. The approximation is granularly representable (without inconsistencies) and as close as possible to the original fuzzy set (w.r.t. a given loss function). It can be seen as a fuzzy counterpart of the monotone approximation produced by the KS approach. As in the work of Kotłowski and Słowiński, we provided statistical foundations of the granular approximations. In the next step, we formulated optimization problems in order to calculate the approximations and we showed some of their important properties. The optimization problems were also considered from the combinatorial perspective. The dual linear programs are formulated as (modified) min-cost flow problems and solved using combinatorial techniques. The solutions are then used to prove Property 6.3.3. At the end, we provided two didactic examples; one for a binary classification problem and one for a regression problem. In the didactic examples, we showed how fuzzy relations are used to model relationships among numerical data, how the granular approximations are calculated and how to interpret them in the two cases for different loss functions.

In the next chapter, the granular properties of the granular approximations are considered. We show that granules form the granular approximations exhibit specific properties (adjacency) which can be used to formulate a multi-class classification version of granular approximation. Machine learning applications of granular approximations will be explored in Chapter 8 where the instance-based classification method will be developed.

Chapter 7

Multi-Class Granular Approximation by means of T-disjoint and Adjacent Fuzzy Granules

In this chapter, we extend the definition of granular approximations to the multi-class context. We introduce the concepts of T-disjoint and adjacent fuzzy granules and we discuss how these concepts relate to the formerly introduced granular approximation. They are important in classification problems since they help us to keep decision regions separated (T-disjoint granules) while covering as much as possible of an attribute space (adjacent granules). Then, we formulate an optimization procedure in order to extend granular approximations to the multi-class classification problem leading to the definition of *multi-class granular approximations*. Such approximation is a union of fuzzy granules constructed in the way described in the previous chapters; it is a fuzzy set constructed as a conjunction of a fuzzy relation and an association value. These association values, as discussed in Chapter 6, can be interpreted as the degree up to which an instance belongs to a certain decision class.

The chapter is structured as follows. Section 7.1 deals with the concept of T-disjoint granules and adjacent granules. It provides definitions of the concepts together with an analysis of how these definitions pertain to the granular approximations introduced in the previous chapter. Section 7.2 explains how the concepts from the previous sections can be applied in binary and multi-class classification problems, and it intro-

duces the definition of a multi-class granular approximation. Section 7.3 shows how to efficiently calculate multi-class granular approximations and provides a graphical illustration of how the granules look in practice. Section 7.4 concludes the chapter.

7.1 T-disjoint and adjacent granules

7.1.1 Definitions and basic properties

Throughout this chapter, we assume that $\widetilde{R}$ is a T-preorder relation for a residual triplet (T, I, N).

Definition 7.1.1. Two fuzzy sets A and B, defined on universe U, are called T-disjoint if

$$T(A(u), B(u)) = 0 \text{ for every } u \in U.$$

In (5.1), the fuzzy granule w.r.t. T-preorder $\widetilde{R}$ was introduced. We now define fuzzy granules w.r.t. inverse relation $\widetilde{R}^{-1}$ as:

$$\widetilde{R}^{-}_{\lambda}(u) = \{(v, T(\widetilde{R}^{-1}(v,u), \lambda)); v \in U\} = \{(v, T(\widetilde{R}(u,v), \lambda)); v \in U\}. \tag{7.1}$$

For the fuzzy granules defined by Eq. (5.1), we have the following property:

Proposition 7.1.1. Let $u, v \in U$. Two fuzzy granules $\widetilde{R}^{+}_{\lambda_1}(u)$ and $\widetilde{R}^{-}_{\lambda_2}(v)$ are T-disjoint if and only if

$$T(\lambda_1, \lambda_2) \leq N(\widetilde{R}(v,u)). \tag{7.2}$$

Proof. The statement that two granules are T-disjoint is equivalent to:

$$\begin{aligned}
&\max_{w \in U} T(T(\widetilde{R}(w,u), \lambda_1), T(\widetilde{R}(v,w), \lambda_2)) = 0 \\
\Leftrightarrow &\max_{w \in U} T(T(\widetilde{R}(v,w), \widetilde{R}(w,u)), T(\lambda_1, \lambda_2)) = 0 \\
\Leftrightarrow &T\left(\max_{w \in U} T(\widetilde{R}(v,w), \widetilde{R}(w,u)), T(\lambda_1, \lambda_2)\right) = 0 \\
\Leftrightarrow &T(\widetilde{R}(v,u), T(\lambda_1, \lambda_2)) = 0 \\
\Leftrightarrow &T(\lambda_1, \lambda_2) \leq I(\widetilde{R}(v,u), 0) \\
\Leftrightarrow &T(\lambda_1, \lambda_2) \leq N(\widetilde{R}(v,u)).
\end{aligned}$$

The first equivalence holds because of the commutativity and associativity of T. The second one holds because T is left-continuous. The third one is a consequence of the T-transitivity of $\widetilde{R}$ while the fourth equivalence follows from the residuation property. □

Please note that the T-disjointness is characterised by the above proposition only for granules of opposite types, i.e., granules w.r.t. relations $\widetilde{R}$ and $\widetilde{R}^{-1}$ respectively.

Proposition 7.1.2. Let (T,I,N) be an IMTL triplet. Then, fuzzy set A is granularly representable w.r.t. $\widetilde{R}$ if and only if the granules from A (w.r.t. $\widetilde{R}$) and coA (w.r.t. $\widetilde{R}^{-1}$) are T-disjoint.

Proof. For $u,v \in U$, we have the following equivalences.

$$\begin{aligned} A(u) \geq T(\widetilde{R}(u,v),A(v)) &\Leftrightarrow \widetilde{R}(u,v) \leq I(A(v),A(u)) \\ &\Leftrightarrow N(\widetilde{R}(u,v)) \geq N(I(A(v),A(u))) \\ &\Leftrightarrow N(\widetilde{R}(u,v)) \geq T(A(v),N(A(u))). \end{aligned}$$

The last equivalence holds because of (2.6g), while the second one follows from the fact that N is a decreasing function. The equivalences state that the granular representability of A is equivalent to the T-disjointness condition of granules $\widetilde{R}^{+}_{A(v)}(v)$ from A and $\widetilde{R}^{-}_{coA(u)}(u)$ from coA, as formulated in Proposition 7.1.1. □

Corollary 7.1.1. Let (T,I,N) be an IMTL triplet. The granules from $\underline{\mathrm{apr}}^{\min,I}_{R}(A)$ and $\overline{\mathrm{apr}}^{\max,T}_{R}(coA)$ are T-disjoint (analogously, the granules from $\underline{\mathrm{apr}}^{\min,I}_{R}(coA)$ and $\overline{\mathrm{apr}}^{\max,T}_{R}(A)$ are T-disjoint too).

Proof. The result holds from the duality property of the lower and upper approximations (4.8). □

Next, we examine a pair of granules $\widetilde{R}^{+}_{\lambda_1}$ and $\widetilde{R}^{-}_{\lambda_2}$ that are not only T-disjoint, but are adjacent to each other. In other words, if their parameters are λ_1 and λ_2, then adding any ϵ to either λ_1 or λ_2 will cause the granules to overlap. For fixed λ_1, the largest λ_2 for which the granules are still T-disjoint is:

$$\lambda_2^{\max} = \sup\{\lambda; T(\lambda_1,\lambda) \leq N(\widetilde{R}(v,u))\} = I(\lambda_1, N(\widetilde{R}(v,u))).$$

Obviously,

$$T(\lambda_1,\lambda_2^{\max}) = T(\lambda_1, I(\lambda_1, N(\widetilde{R}(v,u)))) \leq N(\widetilde{R}(v,u)),$$

due to the modus ponens property (2.6c).

Definition 7.1.2. Granule $\widetilde{R}^{-}_{\lambda_2}(v)$ is adjacent to granule $\widetilde{R}^{+}_{\lambda_1}(u)$ if

$$\lambda_1 = I(\lambda_2, N(\widetilde{R}(v,u))),$$

while $\widetilde{R}^{+}_{\lambda_1}(v)$ is adjacent to $\widetilde{R}^{-}_{\lambda_2}(u)$ if

$$\lambda_2 = I(\lambda_1, N(\widetilde{R}(u,v))).$$

We call such defined relationship among granules the adjacency relation.

Proposition 7.1.3. Every granule is adjacent to all granules with parameter 1, under the assumption that they are T-disjoint.

Proof. If $\lambda_1 = 1$, from the T-disjointness property we have:

$$T(1,\lambda_2) \leq N(\widetilde{R}(v,u)) \Leftrightarrow 1 \leq I(\lambda_2, N(\widetilde{R}(v,u))) \Rightarrow 1 = I(\lambda_2, N(\widetilde{R}(v,u))).$$

□

From the proof of Proposition 7.1.3, we conclude that granule $\widetilde{R}^{+}_{\lambda_1}(u)$ for $\lambda_1 = 1$ is adjacent to $\widetilde{R}^{-}_{\lambda_2}(v)$ if and only if $\lambda_2 = N(\widetilde{R}(v,u))$.

The previous reasoning also reveals that the adjacency relation is not necessarily symmetric.

Proposition 7.1.4. For parameters λ_1 and λ_2 that are smaller than 1 and for continuous t-norm T from the IMTL triplet (T, I, N), we have that the adjacency relation is symmetric. In other words, if $\lambda_1 = I(\lambda_2, N(\widetilde{R}(v,u)))$, then also $\lambda_2 = I(\lambda_1, N(\widetilde{R}(v,u)))$.

Proof. Ordering property (2.6d) implies that if $\lambda_1 < 1$, then also $\lambda_2 > N(\widetilde{R}(v,u))$. Using the strong max-definability (2.8), we have that

$$\lambda_2 = I(I(\lambda_2, N(\widetilde{R}(v,u))), N(\widetilde{R}(v,u))) = I(\lambda_1, N(\widetilde{R}(v,u))).$$

□

Example 7.1.1. Figures 7.1 and 7.2 illustrate different relationships between granules, in one and two dimensions respectively. In Figure 7.1, objects are represented using one condition attribute q whose range is 1. There are two objects u_1 and v_1 with respective attribute values 0.4 and 0.6. Their granules $R^{-}_{\lambda_1}(u_1)$ and $R^{+}_{\lambda_2}(v_1)$ are formed based on a T-preorder relation (left side of the figure) and on a T-equivalence relation

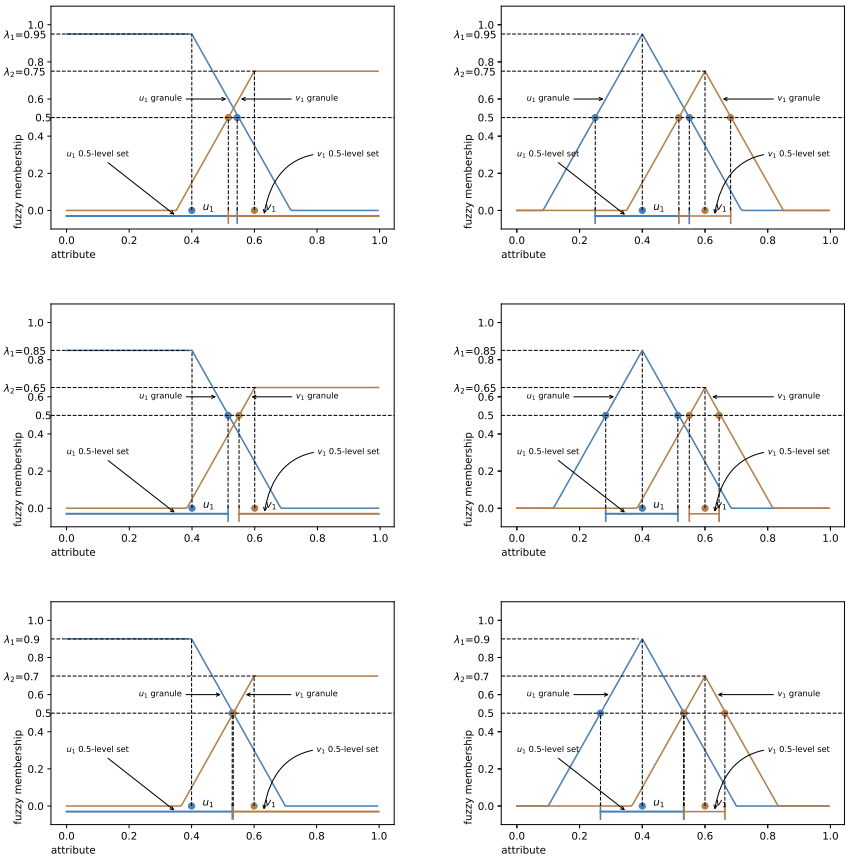

Figure 7.1: Granules in one dimension

(right side of the figure), parameter value $\gamma = 3$ and the Łukasiewicz t-norm. We vary parameters λ_1 and λ_2 in order to represent different relationship among two granules. We depict the fuzzy granules together with their 0.5-level sets. In the upper two images, the values of parameters are $\lambda_1 = 0.95$ and $\lambda_2 = 0.75$ which leads to overlapping granules (i.e., they are not T-disjoint). In the two images in the middle, the values of parameters are $\lambda_1 = 0.85$ and $\lambda_2 = 0.65$ which leads to T-disjoint granules, while in the lower two images, the values of parameters are $\lambda_1 = 0.9$ and $\lambda_2 = 0.7$ which leads to adjacent granules (here, the adjacency relation is symmetric). It is easy to verify that in this case, the 0.5-level sets follow the relation between granules, i.e., if the granules overlap, then the level sets overlap, if the granules are T-disjoint, then

the level sets are disjoint and if the granules are adjacent, then the level sets have one common point.

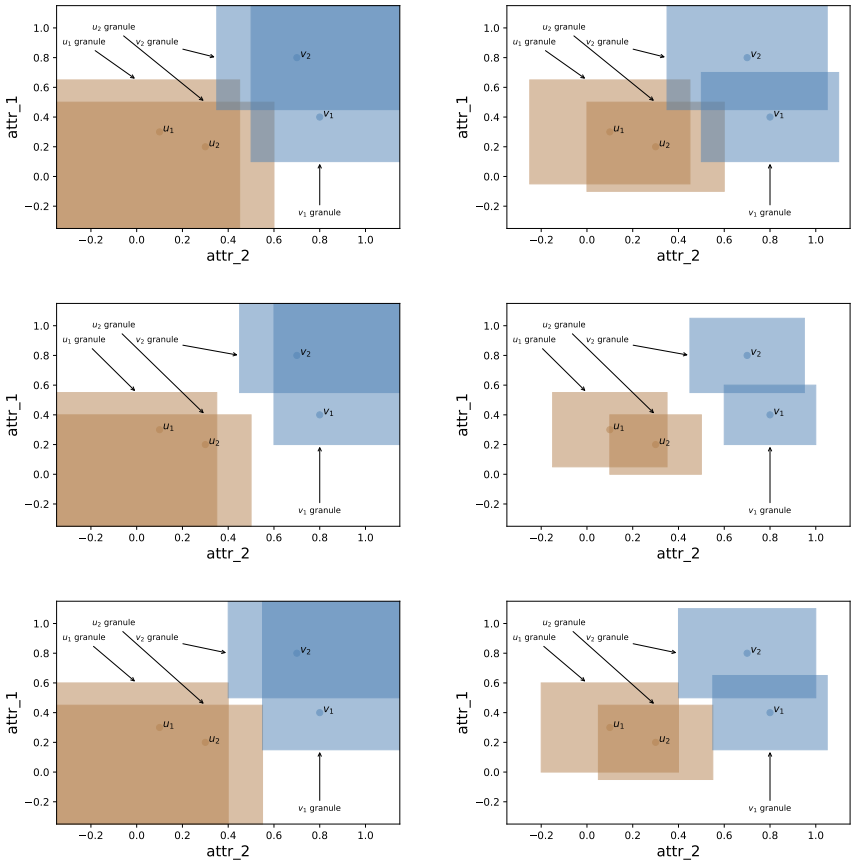

Figure 7.2: Granules in two dimensions

The use of 0.5-level sets is in particular useful to visualize the granules in the case of two dimensions. In Figure 7.2, we have four objects from two classes, described by two condition attributes; u_1 and u_2 are from one class and v_1 and v_2 are from the other one. We illustrate the relationship of granules from different classes. The granules are formed using a T-preorder relation (left side of the figure) and T-equivalence relation (right side of the figure). While the granules in one dimension had triangular (T-equivalence) or "half-triangular" shape (T-preorder), in two dimensions they have pyramidal (T-equivalence) or "half-pyramidal" (T-preorder) shape. However, for the purpose of

the visualisation in 2 dimensions of the 3-dimensional granules, we use 0.5-level sets of the granules. The level sets for the T-preorder relation take the form of quarter-planes (left images) and the rectangular form (right images) for the T-equivalence relation. Again, we can distinguish granules from different classes that overlap (upper two images), that are T-disjoint (two images in the middle) and that are adjacent (lower two images). In this case, the level sets of adjacent granules share an edge.

In Proposition 7.1.2 we showed that granules from A and coA are T-disjoint for granularly representable fuzzy set A. Now we examine in which cases some of them are also adjacent. If $\widetilde{R}^{-}_{coA(v)}(v)$ is adjacent to $\widetilde{R}^{+}_{A(u)}(u)$ for $u, v \in U$, we have that

$$A(u) = I(N(A(v)), N(\widetilde{R}(v,u))) \Leftrightarrow A(u) = I(\widetilde{R}(v,u), A(v)), \tag{7.3}$$

where the equivalence holds because of (2.7a). If $\widetilde{R}^{+}_{A(v)}(v)$ is adjacent to $\widetilde{R}^{-}_{coA(u)}(u)$ then

$$\begin{aligned} N(A(u)) = I(A(v), N(\widetilde{R}(u,v))) &\Leftrightarrow A(u) = N(I(A(v), N(\widetilde{R}(u,v)))) \\ &\Leftrightarrow A(u) = T(\widetilde{R}(u,v), A(v)), \end{aligned} \tag{7.4}$$

where the second equivalence holds because of (2.6h).

7.1.2 Application to granular approximations

In Proposition 7.1.2, we proved that the granules associated with A and coA are T-disjoint if A is granularly representable. In this section, we show that when A is a granular approximation, i.e., a solution of optimization problem (6.1), every granule associated with a particular decision (either A or coA) has at least one granule associated with the opposite decision that is adjacent to it.

In this section, we show that every granule associated with one decision (A or coA) has at least one adjacent granule associated with the opposite decision.

We first recall the notation from Chapter 6 where with $\bar{A}$ we denote the observed values of A (those that are approximated) while with $\hat{A}$ we denote the result of optimization procedure (6.1) (the granular approximation). The next lemma, theorem, and corollary investigate the adjacency relationship of granules from granular approximations, i.e., from the solutions of optimization problem (6.1).

Lemma 7.1.1. Let loss function L be of $\vee$-type and let $\hat{A}$ be a solution of optimization problem (6.1). If $\hat{A}(u) > \bar{A}(u)$, then it holds that

$$\hat{A}(u) = \max\{T(\widetilde{R}(u,v), \hat{A}(v)); v \in U, v \neq u\},$$

while if $\hat{A}(u) < \bar{A}(u)$, then it holds that

$$\hat{A}(u) = \min\{I(\widetilde{R}(v,u), \hat{A}(v)); v \in U, v \neq u\}.$$

Proof. Let $\alpha_u = \max\{T(\widetilde{R}(u,v), \hat{A}(v)); v \in U, v \neq u\}$ for some $u, \hat{A}(u) > \bar{A}(u)$. If the first condition of the theorem is not satisfied, then from the granular representability it holds that $\alpha_u < \hat{A}(u)$. Replacing $\hat{A}(u)$ with $\max(\alpha_u, \bar{A}(u))$ leads to a solution that is also granularly representable (easily verifiable) and that ensures a smaller value of the objective function since the loss function L is of $\vee$-type. That is a contradiction.

Now, let $\alpha_u = \min\{I(\widetilde{R}(v,u), \hat{A}(v)); v \in U, v \neq u\}$ for some $u, \hat{A}(u) < \bar{A}(u)$. If the second condition of the theorem is not satisfied, then from the granular representability it holds that $\alpha_u > \hat{A}(u)$. Replacing $\hat{A}(u)$ with $\min(\alpha_u, \bar{A}(u))$ leads to a solution that is also granularly representable (easily verifiable) and that ensures a smaller value of the objective function since the loss function L is of $\vee$-type. That is a contradiction. □

Theorem 7.1.1. Let loss function L be of $\vee$-type and let $\hat{A}$ be a solution of optimization problem (6.1). We define three sets:

- $U^- = \{u \in U; \hat{A}(u) < \bar{A}(u)\}$,
- $U^0 = \{u \in U; \hat{A}(u) = \bar{A}(u)\}$,
- $U^+ = \{u \in U; \hat{A}(u) > \bar{A}(u)\}$.

It holds that

$$\hat{A}(u) = \max\{T(\widetilde{R}(u,v), \hat{A}(v)); v \in U^- \cup U^0\}, \tag{7.5}$$

for $u \in U^+$ and

$$\hat{A}(u) = \min\{I(\widetilde{R}(v,u), \hat{A}(v)); v \in U^+ \cup U^0\}, \tag{7.6}$$

for $u \in U^-$.

Proof. Condition (7.5) can be reformulated as

$$\forall u \in U^+, \exists v \in U^- \cup U_0; \hat{A}(u) = T(\widetilde{R}(u,v), \hat{A}(v)).$$

Let $U_1^+ \subseteq U^+$ be a set of instances that satisfy the previous condition and let $U_2^+ = U^+ - U_1^+$. Let $u^* \in U_2^+$ be an instance with the largest $\hat{A}(u)$. From Lemma 7.1.1, there is $v \in U, v \neq u^*$ such that $\hat{A}(u^*) = T(\widetilde{R}(u^*,v), \hat{A}(v))$.

By the assumption $u^* \in U_2^+$, we have that $v \in U^+$. If $v \in U_1^+$, then there is $w \in U^- \cup U^0$ such that $\hat{A}(v) = T(\widetilde{R}(v,w), \hat{A}(w))$. We have that

$$\begin{aligned}
\hat{A}(u^*) &= T(\widetilde{R}(u^*,v), \hat{A}(v)) \\
&= T(\widetilde{R}(u^*,v), T(\widetilde{R}(v,w), \hat{A}(w))) \\
&= T(T(\widetilde{R}(u^*,v), \widetilde{R}(v,w)), \hat{A}(w)) \\
&\leq T(\widetilde{R}(u^*,w), \hat{A}(w)).
\end{aligned}$$

The last inequality holds because of the T-transitivity of $\widetilde{R}$ and the monotonicity of T.

The opposite inequality holds from the granular representability which leads to the conclusion that $\hat{A}(u^*) = T(\widetilde{R}(u^*,w), \hat{A}(w))$ which contradicts the assumption that $u^* \in U_2^+$. Hence, $v \in U_2^+$.

From $\hat{A}(u^*) = T(\widetilde{R}(u^*,v), \hat{A}(v))$, it holds $\hat{A}(v) \geq \hat{A}(u^*)$ due to (2.6a). Since $\hat{A}(u^*)$ is the largest in U_2^+ by the assumption, then $\hat{A}(u^*) = \hat{A}(v)$. Denote with $U_3^+ \subseteq U_2^+$ instances from U_2^+ for which the membership degree in $\hat{A}$ is $\hat{A}(u^*)$. Every pair of instances from U_3^+ satisfies (5.2) since they have the same membership value in $\hat{A}$. Due to maximality of $\hat{A}(u)$ it holds that for $u \in U_3^+$ and for $v \in U - U_3^+$, it holds that $\hat{A}(u) > T(\widetilde{R}(u,v), \hat{A}(v))$. Denote

$$\begin{aligned}
\beta^+ = \max(&\max\{\bar{A}(u); u \in U_3^+\}, \\
&\max\{\max\{T(\widetilde{R}(u,v), \hat{A}(v)); v \in U - U_3^+\}; u \in U_3^+\}).
\end{aligned}$$

From the assumptions above, it holds that $\beta^+ < \hat{A}(u^*)$ which implies $\beta^+ < \hat{A}(u)$ for $u \in U_3^+$. Now, let $\hat{A}^*$ be a fuzzy set where values $\hat{A}(u)$ for $u \in U_3^+$ are replaced with β^+. We observe that $\hat{A}^*$ is granularly representable since $\hat{A}^*(u)$ are pairwise equal for $u \in U_3^+$ and also for every $u \in U_3^+$, and for every $v \in U - U_3^+$ it holds that

$$T(\widetilde{R}(u,v), \hat{A}^*(v)) \leq \hat{A}^*(u),$$

by the definition of β^+. Next, we observe that the objective value with $\hat{A}^*$ is smaller than with $\hat{A}$ because

$$\bar{A}(u) < \hat{A}^*(u) < \hat{A}(u),$$

for $u \in U_3^+$ and due to the fact that L is of $\vee$-type.

Therefore, we obtained a feasible solution with a smaller objective function, which contradicts the optimality of $\hat{A}$. This contradiction implies that U_2^+ must be empty, which is equivalent to (7.5).

On the other hand, condition (7.6) can be reformulated as

$$\forall u \in U^-, \exists v \in U^+ \cup U_0; \hat{A}(u) = I(\widetilde{R}(v,u), \hat{A}(v)).$$

Let $U_1^- \subseteq U^-$ be the set of instances that satisfy the previous condition and let $U_2^- = U^- - U_1^-$. Let $u^* \in U_2^-$ be an instance with the smallest $\hat{A}(u)$. From Lemma 7.1.1, there is $v \in U, v \neq u^*$ such that $\hat{A}(u^*) = I(\widetilde{R}(v,u^*), \hat{A}(v))$.

By the assumption $u^* \in U_2^-$, it holds that $v \in U^-$. Assume that $v \in U_1^-$. Then, there is $w \in U^+ \cup U^0$ such that $\hat{A}(v) = I(\widetilde{R}(w,v), \hat{A}(w))$. We have that

$$\begin{aligned}
\hat{A}(u^*) &= I(\widetilde{R}(v,u^*), \hat{A}(v)) \\
&= I(\widetilde{R}(v,u^*), I(\widetilde{R}(w,v), \hat{A}(w))) \\
&= I(T(\widetilde{R}(v,u^*), \widetilde{R}(w,v)), \hat{A}(w)) \\
&\geq I(\widetilde{R}(w,u^*), \hat{A}(w)).
\end{aligned}$$

The last equality holds because of (2.6f), while the last inequality holds because of the T-transitivity of $\widetilde{R}$ and the fact that I is decreasing in its first argument. The opposite inequality holds from the granular representability, which leads to the conclusion that $\hat{A}(u^*) = I(\widetilde{R}(w,u^*), \hat{A}(w))$ which contradicts the assumption that $u^* \in U_2^+$. Because of this, $v \in U_2^-$.

From $\hat{A}(u^*) = I(\widetilde{R}(v,u^*), \hat{A}(v))$, it holds that $\hat{A}(v) \leq \hat{A}(u^*)$ due to (2.6b). Since $\hat{A}(u^*)$ is the smallest by the assumption, then $\hat{A}(u^*) = \hat{A}(v)$. Denote with $U_3^- \subseteq U_2^-$ instances from U_2^- that have value $\hat{A}(u^*)$. Every pair of instances from U_3^- satisfy (5.2) since they have the same membership degree in $\hat{A}$. For every $u \in U_3^-$ and for every $v \in U - U_3^-$ it holds that $\hat{A}(u) < I(\widetilde{R}(v,u), \hat{A}(v))$. Denote

$$\begin{aligned}
\beta^- = \min(&\min\{\bar{A}(u); u \in U_3^-\}, \\
&\min\{\min\{I(\widetilde{R}(v,u), \hat{A}(v)); v \in U - U_3^-\}; u \in U_3^-\}).
\end{aligned}$$

From the above assumption, it holds that $\beta^- > \hat{A}(u^*)$, which implies $\beta > \hat{A}(u)$ for $u \in U_3^-$. Now, let $\hat{A}^{**}$ be a fuzzy set where values $\hat{A}(u)$ for $u \in U_3^-$ are replaced with β^-. We observe that $\hat{A}^{**}$ is granularly representable since $\hat{A}^{**}(u)$ are pairwise equal for $u \in U_3^-$ and also for every $u \in U_3^-$ and for every $v \in U - U_3^-$ it holds that

$$I(\widetilde{R}(v,u), \hat{A}^{**}(v)) \leq \hat{A}^{**}(u),$$

by the definition of β^-. Next, we observe that the objective value with $\hat{A}^{**}$ is smaller than with $\hat{A}$ because

$$\bar{A}(u) > \hat{A}^{**}(u) > \hat{A}(u)$$

for $u \in U_3^-$ and due to the fact that L is of $\vee$-type.

Therefore, we obtained a feasible solution with a smaller objective function, which contradicts the optimality of $\hat{A}$. This contradiction implies that U_2^- must be empty, which is equivalent to (7.6).

□

Coming back to the discussion from the beginning of this section, we formulate the following corollary in order to identify adjacent granules associated to opposite decisions.

Corollary 7.1.2. Let loss function L be of $\vee$-type and let $\hat{A}$ be a solution of optimization problem (6.1) defined w.r.t. an IMTL triplet (T, I, N). Let U^-, U^0, U^+ be defined as in Theorem 7.1.1. Then, the following holds.

- For all $u \in U^+$, there is $v \in U^- \cup U_0$ such that $R^+_{\hat{A}(v)}(v)$ is adjacent to $R^-_{co\hat{A}(u)}(u)$.
- For all $u \in U^-$, there is $v \in U^+ \cup U_0$ such that $R^-_{co\hat{A}(v)}(v)$ is adjacent to $R^+_{\hat{A}(u)}(u)$.

Proof. The corollary is a direct consequence of Theorem 7.1.1 and equations (7.3) and (7.4). □

Note that Theorem 7.1.1 does not require for residual triplet (T, I, N) to be an IMTL triplet, hence it can lead to more general results that are not related to the granular adjacency relationships.

7.2 Case of a classification problem

First, we consider a binary classification problem, i.e., we distinguish two observed decision classes in U: $\bar{A}$ and $co\bar{A}$ which are now crisp (ordinary) sets. Notations $\bar{A}$ and $co\bar{A}$ will also be used for the fuzzy sets that encode the corresponding decision class, i.e., $\bar{A}(u) = 1$ if $u \in \bar{A}$ while $\bar{A}(u) = 0$ if $u \in co\bar{A}$.

We first adjust Theorem 7.1.1 for the binary classification case.

Proposition 7.2.1. Let loss function L be of $\vee$-type and let $\hat{A}$ be a granular approximation of a crisp set A. Then, the following expressions hold:

$$\forall u \in coA, \hat{A}(u) = \max_{w \in \bar{A}} T(\widetilde{R}(u,w), \hat{A}(w)),$$

$$\forall u \in A, \hat{A}(u) = \min_{w \in co\bar{A}} I(\widetilde{R}(w,u), \hat{A}(w)).$$

Proof. Let U^+, U^0, U^- be the sets defined in Theorem 7.1.1. Obviously, it holds that $U^- \subseteq \bar{A}$ and $U^+ \subseteq co\bar{A}$. We prove the first expression, while the second one holds by analogy. The first expression is equivalent to

$$\forall u \in co\bar{A}, \exists v \in \bar{A}; \hat{A}(u) = T(\widetilde{R}(u,v), \hat{A}(v)).$$

Assume that $u \in co\bar{A}$.

If $u \in co\bar{A} - U^+ \Leftrightarrow \hat{A}(u) = 0$, then from the granularity property we have that for all $v \in co\bar{A}$, $\hat{A}(u) \geq T(\widetilde{R}(u,v), \hat{A}(v)) \Rightarrow \hat{A}(u) = T(\widetilde{R}(u,v), \hat{A}(v))$.

If $u \in U^+$, then from Theorem 7.1.1, there is $v \in U^0 \cup U^-$ such that $\hat{A}(u) = T(\widetilde{R}(u,v), \hat{A}(v))$. If $v \in U^-$, then also $v \in \bar{A}$ since $U^- \subseteq \bar{A}$. If $v \in U^0$, then either $\hat{A}(v) = 0$ or $\hat{A}(v) = 1$. If $\hat{A}(v) = 0$ then we have that

$$\hat{A}(u) = T(\widetilde{R}(u,v), \hat{A}(v)) = T(\widetilde{R}(u,v), 0) = 0,$$

which contradicts the assumption that $u \in U^+$. Therefore, it holds that $\hat{A}(v) = 1$, which, combined with the fact that $v \in U^0$, implies $v \in \bar{A}$. □

Corollary 7.2.1. Let loss function L be of $\vee$-type and let $\hat{A}$ be a granular approximation of a crisp set $\bar{A}$ w.r.t. an IMTL triplet (T, I, N). Then, the following holds.

- For all $u \in \bar{A}$, there is $v \in co\bar{A}$ such that $R^-_{co\hat{A}(v)}(v)$ is adjacent to $R^+_{\hat{A}(u)}(u)$.
- For all $u \in co\bar{A}$, there is $v \in \bar{A}$ such that $R^+_{\hat{A}(v)}(v)$ is adjacent to $R^-_{co\hat{A}(u)}(u)$.

Proof. The corollary is a direct consequence of Proposition 7.2.1 and equations (7.3) and (7.4). □

For a solution $\hat{A}$ of optimization problem (6.1), we have that $\hat{A}(u)$ for $u \in U$ represents the degree up to which u belongs to decision class A, while $co\hat{A}(u)$ represents the degree up to which u belongs to decision class coA. Denote $\beta_u = \hat{A}(u)$ for $u \in \bar{A}$ and $\beta_u = N(\hat{A}(u))$ for $u \in co\bar{A}$. We

refer to the new notation as an *alternative notation*. While $\hat{A}(u)$ stands for an estimated membership degree of u in A, β_u is an estimated membership degree of u in the observed class of u (it can be either A or coA if $\bar{A}(u) = 1$ or $\bar{A}(u) = 0$ respectively).

The alternative notation is important to extend the optimization procedure (6.1) to the multi-class classification case.

Proposition 7.2.2. Let L be of $\vee$-type and N-duality preserving and symmetric. Then, in the classification case, problem (6.1) is equivalent to

$$\begin{aligned} \text{minimize} \quad & \sum_{u \in U} L(1, \beta_u) \\ \text{subject to} \quad & T(\beta_u, \beta_v) \leq N(\widetilde{R}(v,u)), \quad u \in \bar{A}, v \in co\bar{A} \\ & 0 \leq \beta_u \leq 1, \quad u \in U. \end{aligned} \tag{7.7}$$

Proof. With the new notation and for L being N-duality preserving, the objective function of (6.1) becomes:

$$\sum_{u \in \bar{A}} L(1, \beta_u) + \sum_{u \in co\bar{A}} L(0, N(\beta_u)) = \sum_{u \in \bar{A}} L(1, \beta_u) + \sum_{u \in co\bar{A}} L(\beta_u, 1) = \sum_{u \in U} L(1, \beta_u).$$

The granularity constraints from (6.1) are now divided into 3 groups:

- Granularity constraints for pairs of objects $u, v \in \bar{A}$:

$$\beta_u \geq T(\widetilde{R}(u,v), \beta_v).$$

- Granularity constraints for pairs of objects $u, v \in co\bar{A}$:

$$N(\beta_u) \geq T(\widetilde{R}(u,v), N(\beta_v)) \Leftrightarrow \beta_v \geq T(\widetilde{R}(v,u), \beta_u).$$

- Granularity constraints for pairs of objects $u \in \bar{A}, v \in co\bar{A}$. In this case, the granularity condition can be expressed using T-disjointness (according to Proposition 7.1.2) as:

$$T(\beta_u, \beta_v) \leq N(\widetilde{R}(v,u)).$$

The goal is to show that the first two groups of constraints are redundant. We first prove that the adjacency from Proposition 7.2.1 still holds in problem (7.7), i.e., for every $u \in \bar{A}$, there is $v \in co\bar{A}$ such that $\beta_u = I(\beta_v, N(\widetilde{R}(v,u)))$ and that for all $v \in co\bar{A}$, there is $u \in \bar{A}$ such that $\beta_v = I(\beta_u, N(\widetilde{R}(v,u)))$. Using the residuation property, we have that

$$T(\beta_u, \beta_v) \leq N(\widetilde{R}(v,u)) \Leftrightarrow \beta_u \leq I(\beta_v, N(\widetilde{R}(v,u))).$$

If for some u and for all v it holds that $\beta_u < I(\beta_v, N(\widetilde{R}(v,u)))$, then there is $\epsilon > 0$ such that replacing β_u with $\beta_u + \epsilon$ leads to a smaller objective function since the loss function is of $\vee$-type. This leads to a contradiction with the assumption that β is optimal. Again, using the residuation property we have

$$T(\beta_u, \beta_v) \leq N(\widetilde{R}(v,u)) \Leftrightarrow \beta_v \leq I(\beta_u, N(\widetilde{R}(v,u))).$$

Using the same arguments as above, we get the second equality.

Next, we prove that the granularity criteria for β_u and β_v for $u, v \in \bar{A}$ are satisfied. Let $w \in co\bar{A}$ such that $\beta_u = I(\beta_w, N(\widetilde{R}(w,u)))$. From the constraints, it holds that $\beta_v \leq I(\beta_w, N(\widetilde{R}(w,v))) \Leftrightarrow \beta_w \leq I(\beta_v, N(\widetilde{R}(w,v)))$. Then, we have that

$$\begin{aligned}
\beta_u &= I(\beta_w, N(\widetilde{R}(w,u))) \\
&\geq I(I(\beta_v, N(\widetilde{R}(w,v))), N(\widetilde{R}(w,u))) \\
&\geq T(\beta_v, I(N(\widetilde{R}(w,v)), N(\widetilde{R}(w,u)))) \\
&= T(\beta_v, I(\widetilde{R}(w,u), \widetilde{R}(w,v))) \\
&\geq T(\beta_v, \widetilde{R}(u,v)),
\end{aligned}$$

which is exactly the granularity condition for β_u and β_v. Now, let $u, v \in co\bar{A}$ and let $w \in \bar{A}$ be such that $\beta_u = I(\beta_w, N(\widetilde{R}(u,w)))$. From the constraints, it holds that $\beta_w \leq I(\beta_v, N(\widetilde{R}(v,w)))$. Using a similar reasoning as above, we conclude that the granularity condition is also satisfied for β_u and β_v when $u, v \in co\bar{A}$. Since the granularity constraints for pairs of objects from the same class are a consequence of the T-disjointness constraints, they can be omitted in the optimization problem. □

The next goal is to extend optimization procedure with alternative notation (7.7) to the ordinary (non-ordinal) multi-class classification. For that purpose, we assume that $\widetilde{R}(u,v)$ is also a symmetric relation, i.e., it is a T-equivalence. In such case, the granules in A and coA are of the same type. We now consider crisp equivalence relation S on U defined as $S(u,v) = 1$ if u and v are from the same decision class, and $S(u,v) = 0$ otherwise. If u and v are from different decision classes then $I(\widetilde{R}(u,v), S(u,v)) = N(\widetilde{R}(u,v))$, while $I(\widetilde{R}(u,v), S(u,v)) = 1$ otherwise. With relation S, the T-disjointness constraints from (7.7) may be reformulated as

$$T(\beta_u, \beta_v) \leq I(\widetilde{R}(u,v), S(u,v))), \quad u, v \in U. \tag{7.8}$$

Here, we need to note that S, as an equivalence relation, can distinguish among more than two decision classes. In other words, S can be used to model multi-class classification problems. Bearing this in mind, a multi-class extension of problem (7.7) can be formulated as:

$$\begin{aligned} \text{minimize} \quad & \sum_{u \in U} L(1, \beta_u) \\ \text{subject to} \quad & T(\beta_u, \beta_v) \leq I(\widetilde{R}(u,v), S(u,v))), \quad u, v \in U \\ & 0 \leq \beta_u \leq 1, \quad u \in U. \end{aligned} \tag{7.9}$$

We name the result of problem (7.9) a *multi-class granular approximation*.

We need to stress that while the binary classification problem (7.7) with a T-preorder relation is suitable for binary monotone classification problems, i.e., classification problems where there exists a monotone relationship between condition attributes and a decision attribute, the problem (7.9) with a T-equivalence relation is suitable for ordinary classification problems i.e., problems where such monotone relationship cannot be inferred.

When we introduced the alternative notation, it was indicated that we interpret β_u as the estimated membership degree of u in the observed decision class of u. This is justified by Proposition 7.2.2 and the equivalence between (7.7) and (6.1). However, we would like to be able to estimate the membership degree of u in every other decision class.

Assume that we have K decision classes denoted with $A_1, \ldots, A_K$. Let $\bar{A}_1, \ldots, \bar{A}_K$ be observed decision classes from U that are pairwise disjoint and for which the union is equal to U. Then, for $u, v \in U$, relation S from (7.9) is defined as $S(u,v) = 1$ if $\exists k \in \{1, \ldots, K\}$ such that $u \in \bar{A}_k \wedge v \in \bar{A}_k$ and $S(u,v) = 0$ otherwise. Let β_u be a solution of (7.9) with such S. We have the following definition.

Definition 7.2.1. The estimated membership degree of object $u \in U$ in decision class A_k, denoted by $\hat{A}_k(u)$, is defined as follows:

$$\hat{A}_k(u) = \begin{cases} \beta_u, & \text{if} \quad u \in \bar{A}_k, \\ \max_{v \in \bar{A}_k} T(\widetilde{R}(u,v), \beta_v) & \text{otherwise.} \end{cases} \tag{7.10}$$

The first case from (7.10) is inferred from the interpretation of β_u. The second case is inferred from the second part of Proposition 7.2.1 together with the first case. In order to better clarify the second case, we observe that Proposition 7.2.1 formulates the relationship between estimated memberships of two instances in a single decision class when

they are observed in two different decision classes. With such formulation, we estimate membership degrees of instances in decision class A_k using membership degrees of instances that were observed in class A_k and expressed trough values β_u.

The previous definition and the possibility to estimate all membership degrees in all decision classes will be important in Chapter 8, where they will be used to develop a multi-class classification model.

7.3 Calculation

In this section, we will use the following shorthand notation: $M(u,v) = I(\widetilde{R}(u,v), S(u,v))$. We start with an important property.

Proposition 7.3.1. Problem (7.9) has a feasible solution.

Proof. We construct a feasible solution. Let $u_1, \ldots, u_n$ be an ordering of objects from U. We apply the following procedure.

1) β_{u_1} is a random value from $[0,1]$.

2) For $1 < i \leq n$, $\beta_{u_i} = \min\{I(\beta_{u_j}, M(u_j,u_i)); j < i\}$.

The adjacency property is obvious from the construction. We have to prove the granularity property. Let u_i and u_k be two objects for which $k < i$. Since $\beta_{u_i} = \min_{j<i} I(\beta_{u_j}, M(u_i,u_j))$, it holds that $\beta_{u_i} \leq I(\beta_{u_k}, M(u_i,u_k))$. From the residuation property, this is equivalent to $T(\beta_{u_i}, \beta_{u_k}) \leq M(u_i,u_k)$. □

For different IMTL fuzzy connectives and for different loss functions L, problem (7.9) may take forms which cannot be efficiently solved in practice. However, we will consider the problem for two symmetric loss function discussed before: absolute error loss (2.11) or squared error loss (2.10) and T isomorphic to the Łukasiewicz t-norm.

For such fuzzy connectives, the constraints of (7.7) are expressed as

$$\begin{aligned} & \varphi^{-1}(\max(\varphi(\beta_u) + \varphi(\beta_v) - 1, 0)) \leq M(u,v) \\ \Leftrightarrow \quad & \max(\varphi(\beta_u) + \varphi(\beta_v) - 1, 0) \leq \varphi(M(u,v)) \\ \Leftrightarrow \quad & \varphi(\beta_u) + \varphi(\beta_v) \leq 1 + \varphi(M(u,v)), \end{aligned}$$

for isomorphism φ and for $u,v \in U$. We introduce new variables $\forall u \in U, \alpha_u = \varphi(\beta_u)$ and $\forall u,v \in U, M_\varphi(u,v) = \varphi(M(u,v))$. With the new notations, the previous constraints may be expressed as

$$\alpha_u + \alpha_v \leq 1 + M_\varphi(u,v).$$

For the scaled absolute error loss $L_{AEL,\varphi}$, the objective function becomes

$$\sum_{u\in U} |\varphi(1) - \varphi(\beta_u)| = |U| - \sum_{u\in U} \alpha_u,$$

which leads to the optimization problem

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{u\in U} \alpha_u \\ \text{subject to} & \alpha_u + \alpha_v \leq 1 + M_\varphi(u,v), \quad u,v \in U \\ & 0 \leq \alpha_u \leq 1, \quad u \in U. \end{array} \tag{7.11}$$

Optimization problem (7.11) can be solved efficiently using linear programming techniques like the simplex method [133].

For the scaled squared error loss $L_{SEL,\varphi}$, the objective function becomes

$$\sum_{u\in U} (\varphi(1) - \varphi(\beta_u))^2 = \sum_{u\in U} (1 - \alpha_u)^2,$$

which leads to the optimization problem

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{u\in U} (1 - \alpha_u)^2 \\ \text{subject to} & \alpha_u + \alpha_v \leq 1 + M_\varphi(u,v), \quad u,v \in U \\ & 0 \leq \alpha_u \leq 1, \quad u \in U. \end{array} \tag{7.12}$$

Optimization problem (7.12) can be solved efficiently using quadratic programming techniques like the simplex method variation for quadratic programming [52].

Example 7.3.1. In Figure 7.3, the objects come from the well-known iris dataset with, in this case, two features (petal length and petal width) and three classes (setosa, versicolor and virginica). The multi-class granular approximation is calculated by solving problem (7.12) for $\widetilde{R}$ defined by Eq. (3.3), and the granules are depicted using the obtained solution. In this figure, we can observe how granules look on a larger scale (in this case 150 objects).

Next, we provide an example with more complex shapes of granules.

Example 7.3.2. It is also easy to verify that the shape of the level sets of granules, used to represent them in 2 dimensions, are in the case of family (3.6) equal to the shape of equidistant points from the origin w.r.t. metric d. In the case of the Mahalanobis distance, the shape of granules

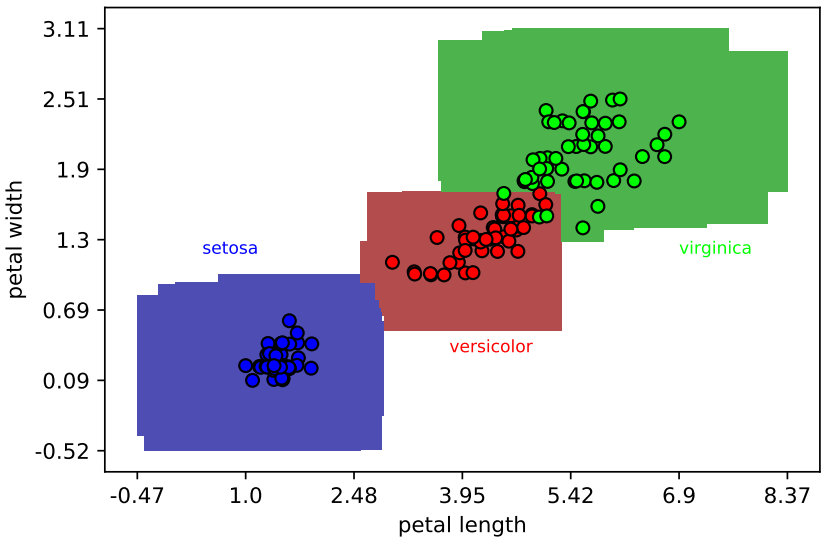

Figure 7.3: An example of the multi-class granular approximation on iris dataset constructed with relation (3.3)

will be elliptical. The axis of such ellipses is controlled by the eigenvalues of Σ while the rotation is controlled by the eigenvectors of Σ.

In Figure 7.4, we present an example of granules from the multi-class granular approximation calculated by solving (7.12) and by using fuzzy relation (3.6) with d the Mahalanobis distance. The approximation is calculated on the iris dataset with two attributes and three decision classes as described above. The granules have an elliptical shape where the ratio of width and height of the ellipses is $2:1$. The rotation angle in this case is $45°$.

In Figures 7.3 and 7.4, we can observe that some green points are depicted without their granules and are completely surrounded by the granules of red points. This basically means that the multi-class granular approximation values of these green points are smaller than 0.5 (hence, the granules cannot be drawn), and that the red granules are covering those green points. In other words, the estimated memberships degrees of those instances in class "versicolor" are larger than the estimated membership degrees in class "virginica". Therefore, it is suitable to change the labels of those green points into red. We can conclude that the learning, characterized by optimization problems (7.11) and (7.12),

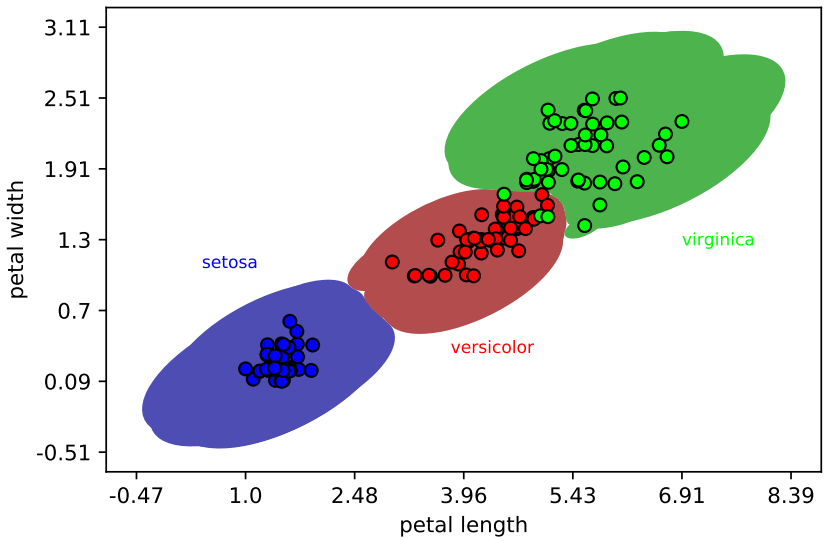

Figure 7.4: An example of the multi-class granular approximation on iris dataset constructed with relation (3.6)

.

can be applied in classification problems as will be later examined in Chapter 8.

7.4 Conclusion

We introduced the concepts of T-disjoint and adjacent fuzzy granules and discussed their connection with the concepts of granularly representable fuzzy sets and granular approximations introduced in the previous chapter. We showed that granules from a granularly representable fuzzy set and its complement are mutually T-disjoint. Moreover, each granule from a granular approximation has an adjacent granule from the complement. Based on this property of granules, a granular approximation concept was applied to the multi-class classification problem leading to the definition of a multi-class granular approximation. At the end, we explained how to calculate it efficiently in practice for the Łukasiewicz t-norm and the other fuzzy connectives that it generates, using linear and quadratic programming methods. In the next chapter, the multi-class granular approximation is used to develop an instance-

based classification model.

Chapter 8

Fuzzy Granular Approximation Classifier

The goal of this chapter is to extrapolate the granular approximation to new, unseen data. We design a classifier that estimates the membership degree of a new instance in a given decision class based on the consistency property. The name of the new classifier is *Fuzzy Granular Approximation Classifier - FGAC*. The classifier is able to perform binary classification as well as multi-class classification natively. It belongs to the family of instance-based classifiers since the prediction is made based on the comparison of a new instance with those from the training set.

The main advantage of the classifier is its interpretability which resides in the following two properties:

- The explanation of the classifier can be derived from the ability to translate fuzzy logic into linguistic expressions.
- It is possible to identify training instances that serve as arguments being in favour or against the prediction, as well as the strength of these arguments.

The new model belongs to the family of locally interpretable models discussed in Section 1.4. For this family of models, we are able to identify how a particular prediction was made. In this chapter, we compare the prediction performance of FGAC with other locally interpretable models discussed in the same section, and we show what are the advantages and disadvantages regarding the interpretability of the FGAC compared to these models. We also provide a brief discussion on the difference

between the interpretability of FGAC and methods that are used to interpret black-box models.

The remainder of this chapter is structured as follows. In Section 8.1, we discuss the required preliminaries for understanding this chapter. In Section 8.2, the novel Fuzzy Granular Approximation Classifier (FGAC) is introduced together with a version enhanced with OWA operators. Section 8.3 contains empirical comparisons between different versions of FGAC and comparisons of FGAC with other ML models. In Section 8.4, we explain why we consider FGAC as an interpretable model and identify its advantages and disadvantages in terms of interpretability compared with other locally interpretable ML models. Section 8.5 concludes the chapter.

8.1 Preliminaries

8.1.1 Datasets

We present the datasets that will be used in the experiments. We collected 23 classification datasets that are available in the UCI Machine Learning repository [40]. Their description is provided in Table 8.1.

name	# of instances	# of numerical attributes	# of nominal attributes	# of classes	distribution of instances among classes
australian	690	8	6	2	(383, 307)
balance	625	4	0	3	(288, 288, 49)
breast	277	0	9	2	(196, 81)
bupa	345	6	0	2	(200, 145)
cleveland	297	13	0	5	(160, 54, 35, 35, 13)
crx	653	6	9	2	(357, 296)
german	1000	7	13	2	(700, 300)
glass	214	9	0	6	(76, 70, 29, 17, 13, 9)
haberman	306	3	0	2	(225, 81)
heart	270	13	0	2	(150, 120)
ionosphere	351	33	0	2	(225, 126)
mammographic	830	5	0	2	(427, 403)
pima	768	8	0	2	(500, 268)
saheart	462	8	1	2	(302, 160)
spectfheart	267	44	0	2	(212, 55)
vowel	990	13	0	11	(90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90)
wdbc	569	30	0	2	(357, 212)
wisconsin	683	9	0	2	(444, 239)
ecoli	336	7	0	8	(143, 77, 2, 2, 35, 20, 5, 52)
dermatology	358	34	0	6	(111, 60, 71, 48, 48, 20)
tic-tac-toe	958	0	9	2	(332, 626)
vehicle	846	18	0	4	(218, 212, 217, 199)
sonar	208	60	0	2	(111, 97)

Table 8.1: Description of datasets

8.1.2 Methods to compare with

The performance of FGAC will be compared with other (up to some degree) locally interpretable classification models that were discussed in Section 1.4.2. These methods are k-Nearest Neigbours [46], k-Fuzzy Rough Nearest Neighbours [78, 110], Classification and Regression Tree [109] and Learning Vector Quantization [87].

k-Nearest Neigbours (kNN) is a non-parametric lazy approach where the decision for a new particular instance is obtained based on the majority decisions of the k closest instances w.r.t. a given distance metric. The interpretability of this approach boils down to our ability to detect the instances based on which the decision was made. However, the interpretability fades as k increases because it becomes hard to understand how a prediction was made based on a high number of other instances.

Classification And Regression Tree (CART) can be seen as a hierarchical rule-based model. In every step of the training phase, a split of the training set of instances is performed based on a provided criterion. In the first step, the whole set of instances is split, while in every subsequent step, a subset of the previous split is chosen and split. This way of splitting creates a binary decision tree. Since every split is performed on a specific attribute, a hierarchical set of rules can be induced in order to explain a particular prediction made by CART. These rules enable the interpretability of the model.

Learning Vector Quantization (LVQ) is a prototype based model, where for each decision class a few points from the attribute space called *prototypes* are learned. These prototypes do not necessarily coincide with the training instances. After the prototypes are learned, a new instance is classified based on the decision of the nearest prototype. The interpretability of LVQ lies in the fact that one is able to identify the prototype responsible for the prediction.

k-Fuzzy Rough Nearest Neighbour (kFRNN) is a lazy approach were for every new instance and for every decision class, we calculate its fuzzy rough lower approximation degree, upper approximation degree and take the mean as the membership degree in that decision class. Then, the decision class is determined as the one for which the highest membership degree is achieved. kFRNN also invokes OWA operators as a replacement for min and max operators in the lower and upper approximations.

8.1.3 Data preprocessing

First, we notice that some datasets from Table 8.1 have nominal features which have to be encoded into numerical ones. For that purpose, one hot encoding is used [51].

We already saw a kind of prepossessing data when defining the triangular similarity in Eq. (3.2) where we divide the values of attribute q by $range(q)$. In this way, we ensure that the largest absolute difference of values within one attribute is 1. Looking from the joint perspective of all attributes, after dividing all of instances by $range(q)$ for the corresponding q, we translate all instances to a unit cube where the largest Chebyshev distance, on which the T-equivalence is based in this case, is equal to 1. With this transformation, we ensure that all attributes contribute to the similarity equally and that $\gamma = 1$ is the default parameter where the T-equivalence is equal to 0 only for the most distant instances. However, we add one practical adaptation here. The most distant instances can be outliers, i.e., they do not necessarily follow the distribution of the data and hence can be misleading in evaluating the proper range. Because of that, as a more robust estimation of $range(q)$, we will use the difference between the .99 quantile, (very close to the maximum) and the .01 quantile (very close to the minimum) to scale the data. This transformation will be applied only when the T-equivalence based on Chebyshev distance, i.e., the triangular similarity, is used.

Beside the Chebyshev distance, the T-equivalence based on the Euclidean distance will also be used. As before where we scaled the instances into a unit cube where the largest Chebyshev distance is one, here we want to scale them such that the largest Euclidean distance will be 1. We do that by dividing the values of each attribute by the standard deviation of that attribute's values. This is done in order to ensure an equal contribution of each attribute to the T-equivalence. Then, the maximum Euclidean distance between instances is calculated and all the instances in all attributes are divided by that value. In this way, we ensure that the largest possible distance is 1. As before, there is a possibility that the largest distance is achieved for some outliers. Therefore, we approximate the largest distance by a high quantile i.e., we calculate all pairwise distances among instances, and we take the .99 quantile as the approximation of the largest distance. This preprocessing will be applied to all methods that are distance based including kNN, kFRNN and LVQ.

8.2 Prediction for unseen objects

In this section, we discuss how to classify a set of unseen instances $U^\dagger$ using optimization problem (6.1).

8.2.1 Binary classification

In this case, we need to assign a membership degree in set A to instances from $U^\dagger$, where A refers to one of the classes. Solving optimization procedure (6.1) does not return an explicit prediction function $f : U \to [0,1]$ which would assign a membership degree to any new and unseen instance from $U^\dagger$. However, the membership degree of any new instance has to satisfy the constraints from (6.1).

Let $u^\dagger \in U^\dagger$. The aim is to estimate the membership degree $\hat{A}(u^\dagger)$. Since unseen objects are represented with condition attributes, the values $\tilde{R}(u^\dagger, u)$ and $\tilde{R}(u, u^\dagger)$ can be calculated for all $u \in U$ and therefore, we assume that they are known. From the constraints of (6.1), we conclude that the conditions:

$$\forall u \in U;\ T(\widetilde{R}(u^\dagger, u), \hat{A}(u)) \leq \hat{A}(u^\dagger),$$

and

$$\forall u \in U;\ T(\widetilde{R}(u, u^\dagger), \hat{A}(u^\dagger)) \leq \hat{A}(u) \Leftrightarrow \forall u \in U;\ \hat{A}(u^\dagger) \leq I(\widetilde{R}(u, u^\dagger), \hat{A}(u)),$$

have to be satisfied. The previous conditions can be rewritten as:

$$\max_{u \in U} T(\widetilde{R}(u^\dagger, u), \hat{A}(u)) \leq \hat{A}(u^\dagger) \leq \min_{u \in U} I(\widetilde{R}(u, u^\dagger), \hat{A}(u)). \tag{8.1}$$

Expression (8.1) determines a lower and an upper bound for membership degree $\hat{A}(u^\dagger)$ which forms an interval to which the degree should belong. First, we have to show that the interval is well defined.

Proposition 8.2.1. For any $u^\dagger \in U^\dagger$, it holds that

$$\max_{u \in U} T(\widetilde{R}(u^\dagger, u), \hat{A}(u)) \leq \min_{u \in U} I(\widetilde{R}(u, u^\dagger), \hat{A}(u)).$$

Proof. An equivalent formulation of the demonstrandum is:

$$\forall u, v \in U,\ T(\widetilde{R}(u^\dagger, u), \hat{A}(u)) \leq I(\widetilde{R}(v, u^\dagger), \hat{A}(v)). \tag{8.2}$$

Using granular representability, T-transitivity and associativity of T, we have that

$$\hat{A}(v) \geq T(\widetilde{R}(v, u), \hat{A}(u))$$

$$\geq T(T(\widetilde{R}(v,u^\dagger),\widetilde{R}(u^\dagger,u)),\hat{A}(u))$$
$$= T(\widetilde{R}(v,u^\dagger),T(\widetilde{R}(u^\dagger,u),\hat{A}(u)).$$

The latter is equivalent to the formulation of the proposition due to the residuation property. □

Since the interval is well defined, the next step is to properly aggregate the lower and upper bounds into a single value. Denote

$$\underline{\hat{A}}(u^\dagger) = \max_{u\in U} T(\widetilde{R}(u^\dagger,u),\hat{A}(u)), \quad \overline{\hat{A}}(u^\dagger) = \min_{u\in U} I(\widetilde{R}(u,u^\dagger),\hat{A}(u)). \tag{8.3}$$

Let $\mathbb{A}$ be an averaging operator. We construct the prediction of the membership degree of $u^\dagger \in A$ as

$$\hat{A}(u^\dagger) = \mathbb{A}(\underline{\hat{A}}(u^\dagger),\overline{\hat{A}}(u^\dagger)). \tag{8.4}$$

The next question is how to construct the averaging operator $\mathbb{A}$. The following development holds for IMTL triplets since it is based on the duality property. Since $\hat{A}(u^\dagger)$ represents the predicted membership degree of $u^\dagger$ to A, then $N(\hat{A}(u^\dagger))$ represents the membership degree to coA. If (8.4) holds, then some sort of duality should also hold, i.e.,

$$N(\hat{A}(u^\dagger)) = \mathbb{A}(co\underline{\hat{A}}(u^\dagger),co\overline{\hat{A}}(u^\dagger)), \tag{8.5}$$

where

$$co\underline{\hat{A}}(u^\dagger) = \max_{u\in U} T(\widetilde{R}(u,u^\dagger),N(\hat{A}(u))),$$
$$co\overline{\hat{A}}(u^\dagger) = \min_{u\in U} I(\widetilde{R}(u^\dagger,u),N(\hat{A}(u))).$$

We have the following result.

Proposition 8.2.2. For every $u^\dagger \in U^\dagger$, it holds that

$$co\overline{\hat{A}}(u^\dagger) = N(\underline{\hat{A}}(u^\dagger)), \quad co\underline{\hat{A}}(u^\dagger) = N(\overline{\hat{A}}(u^\dagger)).$$

Proof. For the left equality, we have that

$$N(\underline{\hat{A}}(u^\dagger)) = N(\max_{u\in U} T(\widetilde{R}(u^\dagger,u),\hat{A}(u)))$$
$$= \min_{u\in U} N(T(\widetilde{R}(u^\dagger,u),\hat{A}(u)))$$
$$= \min_{u\in U} I(\widetilde{R}(u^\dagger,u),N(\hat{A}(u))) = co\overline{\hat{A}}(u^\dagger).$$

The third equality holds from (2.6h). For the right equality, we have that

$$\begin{aligned} N(\overline{\hat{A}}(u^\dagger)) &= N(\min_{u\in U} I(\widetilde{R}(u,u^\dagger),\hat{A}(u))) \\ &= \max_{u\in U} N(I(\widetilde{R}(u,u^\dagger),\hat{A}(u))) \\ &= \max_{u\in U} T(\widetilde{R}(u,u^\dagger),N(\hat{A}(u))) = co\underline{\hat{A}}(u^\dagger). \end{aligned}$$

The third equality holds from (2.7b). □

Following Proposition 8.2.2, we conclude that for an aggregation operator $\mathbb{A}$, it should hold that $N(\hat{A}(u^\dagger)) = \mathbb{A}(N(\underline{\hat{A}}(u^\dagger)), N(\overline{\hat{A}}(u^\dagger)))$, i.e., it is sufficient that $\mathbb{A}$ is N-invariant.

For an involutive negator N, let φ_N be an isomorphism between N and N_s, i.e., $N = \varphi_N^{-1}(N_s(\varphi_N))$. We define an averaging operator:

$$\mathbb{A}_N(x,y) = \varphi_N^{-1}\left(\frac{\varphi_N(x)+\varphi_N(y)}{2}\right). \tag{8.6}$$

It is easily verifiable that $\mathbb{A}_N$ is indeed N-invariant.

Therefore, we predict the membership degree of $u^\dagger$ as

$$\hat{A}(u^\dagger) = \mathbb{A}_N(\underline{\hat{A}}(u^\dagger), \overline{\hat{A}}(u^\dagger)). \tag{8.7}$$

We first want to verify that the predicted membership degrees will satisfy the consistency property. We can show that for the Łukasiewicz triplet $(T_{L,\varphi}, I_{L,\varphi}, N_{L,\varphi})$.

Proposition 8.2.3. Let $u^\dagger, v^\dagger \in U^\dagger$ and let $\hat{A}(u^\dagger)$ and $\hat{A}(v^\dagger)$ be the predicted membership degrees obtained with IMTL triplet $(T_{L,\varphi}, I_{L,\varphi}, N_{L,\varphi})$ and (8.7). Then, it holds that

$$T_{L,\varphi}(\widetilde{R}(u^\dagger,v^\dagger),\hat{A}(v^\dagger)) \le \hat{A}(u^\dagger).$$

Proof. The expression from the proposition is equivalent to:

$$T_L(\widetilde{R}_\varphi(u^\dagger,v^\dagger),\varphi(\hat{A}(v^\dagger))) \le \varphi(\hat{A}(u^\dagger)).$$

We have that:

$$T_L(\widetilde{R}_\varphi(u^\dagger,v^\dagger),\varphi(\hat{A}(v^\dagger))) = T_L\left(\widetilde{R}_\varphi(u^\dagger,v^\dagger), \frac{\varphi(\underline{\hat{A}}(v^\dagger))+\varphi(\overline{\hat{A}}(v^\dagger))}{2}\right)$$

$$\leq \frac{T_L(\widetilde{R}_\varphi(u^\dagger, v^\dagger), \varphi(\underline{\hat{A}}(v^\dagger))) + T_L(\widetilde{R}_\varphi(u^\dagger, v^\dagger), \varphi(\overline{\hat{A}}(v^\dagger)))}{2},$$

where the inequality holds from the D-convexity of T_L. For the first summand in the numerator of the last ratio, we have that

$$\begin{aligned}
T_{L,\varphi}(\widetilde{R}(u^\dagger, v^\dagger), \underline{\hat{A}}(v^\dagger))) &= T_{L,\varphi}(\widetilde{R}(u^\dagger, v^\dagger), \max_{u\in U} T_{L,\varphi}(\widetilde{R}(v^\dagger, u), \hat{A}(u))) \\
&= \max_{u\in U} T_{L,\varphi}(\widetilde{R}(u^\dagger, v^\dagger), T_{L,\varphi}(\widetilde{R}(v^\dagger, u), \hat{A}(u))) \\
&= \max_{u\in U} T_{L,\varphi}(T_{L,\varphi}(\widetilde{R}(u^\dagger, v^\dagger), \widetilde{R}(v^\dagger, u)), \hat{A}(u)) \\
&\leq \max_{u\in U} T_{L,\varphi}(\widetilde{R}(u^\dagger, u), \hat{A}(u)) = \underline{\hat{A}}(u^\dagger)).
\end{aligned}$$

The second equality holds from the fact that $T_{L,\varphi}$ is left-continuous, while the third one from the associativity of the t-norm. The inequality is a consequence of the $T_{L,\varphi}$-transitivity of $\widetilde{R}$. After applying φ to both sides, we obtain $T_L(\widetilde{R}_\varphi(u^\dagger, v^\dagger), \varphi(\underline{\hat{A}}(v^\dagger))) \leq \varphi(\underline{\hat{A}}(u^\dagger))$. For the second summand we have that

$$\begin{aligned}
T_{L,\varphi}(\widetilde{R}_(u^\dagger, v^\dagger), \overline{\hat{A}}(v^\dagger))) &= T_{L,\varphi}(\widetilde{R}(u^\dagger, v^\dagger), \min_{u\in U} I_{L,\varphi}(\widetilde{R}(u, v^\dagger), \hat{A}(u))) \\
&\leq \min_{u\in U} T_{L,\varphi}(\widetilde{R}(u^\dagger, v^\dagger), I_{L,\varphi}(\widetilde{R}(u, v^\dagger), \hat{A}(u))) \\
&= \min_{u\in U} I_{L,\varphi}(I_{L,\varphi}(\widetilde{R}(u^\dagger, v^\dagger), \widetilde{R}(u, v^\dagger)), \hat{A}(u)) \\
&\leq \min_{u\in U} I_{L,\varphi}(\widetilde{R}(u, u^\dagger), \hat{A}(u)) = \overline{\hat{A}}(u^\dagger)).
\end{aligned}$$

The first inequality holds from the fact that $T_{L,\varphi}$ is non-increasing, while the second equality holds from Property (2.6e). The second inequality is a consequence of the residuation property applied on $T_{L,\varphi}$-transitivity of $\widetilde{R}$ (check the proof of Proposition 4.2.4). After applying φ to both sides we obtain $T_L(\widetilde{R}_\varphi(u^\dagger, v^\dagger), \varphi(\overline{\hat{A}}(v^\dagger))) \leq \varphi(\overline{\hat{A}}(u^\dagger))$.

Using the obtained inequality, we have that

$$\begin{aligned}
&\frac{T_L(\widetilde{R}_\varphi(u^\dagger, v^\dagger), \varphi(\underline{\hat{A}}(v^\dagger))) + T_L(\widetilde{R}_\varphi(u^\dagger, v^\dagger), \varphi(\overline{\hat{A}}(v^\dagger)))}{2} \\
\leq &\frac{\varphi(\underline{\hat{A}}(u^\dagger)) + \varphi(\overline{\hat{A}}(u^\dagger))}{2} = \varphi(\hat{A}(u^\dagger)).
\end{aligned}$$

□

After obtaining the predicted membership degree, we have to defuzzify it, i.e., to obtain a crisp binary prediction. We return prediction 1, i.e., $u^\dagger$ belongs to decision A if $\hat{A}(u^\dagger) > N(\hat{A}(u^\dagger))$, and prediction 0 otherwise. Please note that when $\hat{A}(u^\dagger) = N(\hat{A}(u^\dagger))$, we have a tie and any prediction can be assigned. However, we will assign prediction 0 in order to keep the deterministic nature of the prediction model. The condition $\hat{A}(u^\dagger) > N(\hat{A}(u^\dagger))$ can be rewritten as

$$\begin{aligned}\hat{A}(u^\dagger) > N(\hat{A}(u^\dagger)) &\Leftrightarrow \hat{A}(u^\dagger) > \varphi_N^{-1}(1-\varphi_N(\hat{A}(u^\dagger)))\\ &\Leftrightarrow \varphi_N(\hat{A}(u^\dagger)) > 1-\varphi_N(\hat{A}(u^\dagger))\\ &\Leftrightarrow \varphi_N(\hat{A}(u^\dagger)) > \frac{1}{2} \Leftrightarrow \hat{A}(u^\dagger) > \varphi_N^{-1}(0.5).\end{aligned}$$

We obtain that value $\varphi_N^{-1}(0.5)$ is the threshold that determines the decision.

In order to speed up the calculation, we can use the following proposition.

Proposition 8.2.4. In the binary classification case, it holds that

$$\underline{\hat{A}}(u^\dagger) = \max_{u\in\bar{A}} T(\widetilde{R}(u^\dagger,u),\hat{A}(u)), \quad \overline{\hat{A}}(u^\dagger) = \min_{u\in co\bar{A}} I(\widetilde{R}(u,u^\dagger),\hat{A}(u)). \tag{8.8}$$

Proof. An equivalent formulation of the demonstrandum, which holds from the granularity property, is

$$\exists u\in\bar{A};\, \underline{\hat{A}}(u^\dagger) = T(\widetilde{R}(u^\dagger,u),\hat{A}(u)), \quad \exists u\in co\bar{A};\, \overline{\hat{A}}(u^\dagger) = I(\widetilde{R}(u,u^\dagger),\hat{A}(u)). \tag{8.9}$$

We prove the first equality from (8.9). If the maximum from (8.3) is achieved for some $u\in\bar{A}$, the equality is true. Otherwise, we assume that for some $u\in co\bar{A}$, it holds that

$$\underline{\hat{A}}(u^\dagger) = T(\widetilde{R}(u^\dagger,u),\hat{A}(u)).$$

From Proposition 7.2.1, there exists some $v\in\bar{A}$ such that $\hat{A}(u) = T(\widetilde{R}(u,v),\bar{A}(v))$. We have that

$$\begin{aligned}\underline{\hat{A}}(u^\dagger) &= T(\widetilde{R}(u^\dagger,u),\hat{A}(u))\\ &= T(\widetilde{R}(u^\dagger,u),T(\widetilde{R}(u,v),\bar{A}(v)))\\ &= T(T(\widetilde{R}(u^\dagger,u),\widetilde{R}(u,v)),\bar{A}(v))\\ &\leq T(\widetilde{R}(u^\dagger,v),\bar{A}(v)).\end{aligned}$$

The inequality holds from the T-transitivity property. The opposite inequality holds from the granularity property.

For the second inequality from (8.9), assume that the minimum from (8.3) is achieved for some $u \in \bar{A}$. It holds that

$$\overline{\hat{A}}(u^\dagger) = I(\widetilde{R}(u, u^\dagger), \hat{A}(u)).$$

From Proposition 7.2.1, we find that there exists some $v \in co\bar{A}$ such that $\hat{A}(u) = I(\widetilde{R}(v, u), \bar{A}(v))$. We have that

$$\begin{aligned}
\overline{\hat{A}}(u^\dagger) &= I(\widetilde{R}(u, u^\dagger), \hat{A}(u)) \\
&= I(\widetilde{R}(u, u^\dagger), I(\widetilde{R}(v, u), \bar{A}(v))) \\
&= I(T(\widetilde{R}(v, u), \widetilde{R}(u, u^\dagger)), \bar{A}(v)) \\
&\geq I(\widetilde{R}(v, u^\dagger), \bar{A}(v)).
\end{aligned}$$

The third equality holds because of (2.6f), while the inequality follows from the T-transitivity property. The opposite inequality holds from the granularity property which completes the proof. □

8.2.2 Multi-class classification

For the multi-class classification case, we recall the notation from Chapter 7 where we have K decision classes denoted with $A_1, \ldots, A_K$. Using optimization procedure (7.9) and Eq. (7.10) we obtain estimated membership degrees $\hat{A}_k(u)$ for each training instance $u \in U$ and each decision class k.

Using the same reasoning as for binary classification, for $k \in \{1, \ldots, K\}$ and using Proposition 8.2.4, a lower and upper bound of a membership degree of $u^\dagger$ in A_k is obtained as

$$\underline{\hat{A}}_k(u^\dagger) = \max_{u \in \bar{A}_k} T(\widetilde{R}(u^\dagger, u), \hat{A}_k(u)), \quad \overline{\hat{A}}_k(u^\dagger) = \min_{u \in U - \bar{A}_k} I(\widetilde{R}(u, u^\dagger)), \hat{A}_k(u)),$$

while the prediction of the membership degree is obtained using averaging operator (8.6). The decision class is then determined using formula:

$$decision(u^\dagger) = \operatorname*{argmax}_{k \in \{1, \ldots, K\}} \mathbb{A}(\underline{\hat{A}}_k(u^\dagger), \overline{\hat{A}}_k(u^\dagger)).$$

8.2.3 Soft minimum and maximum

From Eqs. (8.1) and (8.4) we observe that the prediction of the membership degree of $u^\dagger$ is obtained based on the extreme values, i.e., the maximum from the left inequality and the minimum from the right inequality. In order to utilize less extreme values, we replace max and min with OWA operators. One motivation for using OWA operators and softening minimum and maximum in general is to reduce the influence of possible outliers in the dataset. The extreme values may correspond to outliers which make the predicted membership degree unreliable. Hence, we wish to explore if using OWA operators will increase the performance of the classification model.

For given weight vectors W_L and W_U that correspond to soft min and max operators respectively, we have the following definitions:

$$\underline{\hat{A}}^{W_U}(u^\dagger) = OWA_{W_U}\{T(\widetilde{R}(u^\dagger, u), \hat{A}(u)); u \in U\},$$
$$\overline{\hat{A}}^{W_L}(u^\dagger) = OWA_{W_L}\{I(\widetilde{R}(u, u^\dagger), \hat{A}(u)); u \in U\},$$

while the estimated membership is obtained in the same way as in Eq. (8.4). From the definition of OWA, for all $u^\dagger \in U^\dagger$ it holds that

$$\underline{\hat{A}}^{W_U}(u^\dagger) \leq \underline{\hat{A}}(u^\dagger), \quad \overline{\hat{A}}^{W_L}(u^\dagger) \geq \overline{\hat{A}}(u^\dagger),$$

which further implies that $\underline{\hat{A}}^{W_U}(u^\dagger) \leq \overline{\hat{A}}^{W_L}(u^\dagger)$, i.e., the bounds are well-defined.

The next question is if the duality expressed in analogous form as in Eq. (8.5) and for N-invariant averaging operator $\mathbb{A}$ will hold for

$$co\underline{\hat{A}}^{W_U}(u^\dagger) = OWA_{W_U}\{T(\widetilde{R}(u, u^\dagger), N(\hat{A}(u))); u \in U\},$$
$$co\overline{\hat{A}}^{W_L}(u^\dagger) = OWA_{W_L}\{I(\widetilde{R}(u^\dagger, u), N(\hat{A}(u))); u \in U\}.$$

If we consider the proof of Proposition 8.2.2, we conclude that the answer to the previous question depends on whether OWA operators and negator N are interchangeable. This is not always the case, but we do have the following proposition.

Proposition 8.2.5.

Let (T, I, N) be a residual triplet for which N is the standard negator and let W_U and W_L be complementary vectors of weights. Then, it holds that

$$co\overline{\hat{A}}^{W_L}(u^\dagger) = N(\underline{\hat{A}}^{W_U}(u^\dagger)), \quad co\underline{\hat{A}}^{W_U}(u^\dagger) = N(\overline{\hat{A}}^{W_L}(u^\dagger)).$$

Proof. We prove the first equality, while the second one holds by analogy. Let $u_1, \ldots, u_n$ be an ordering of instances from U such that

$$T(\widetilde{R}(u^\dagger, u_1), \hat{A}(u_1)) \geq \cdots \geq T(\widetilde{R}(u^\dagger, u_n), \hat{A}(u_n)).$$

Applying negator N to the previous inequalities and using the fact that N is decreasing, together with property (2.6h), we have that

$$I(\widetilde{R}(u^\dagger, u_1), N(\hat{A}(u_1))) \leq \cdots \leq I(\widetilde{R}(u^\dagger, u_n), N(\hat{A}(u_n))).$$

Also,

$$\begin{aligned}
N(\underline{\hat{A}}^{W_U}(u^\dagger)) &= 1 - \sum_{u=1}^{n} (W_U)_i \cdot T(\widetilde{R}(u^\dagger, u_i), \hat{A}(u_i))) \\
&= \sum_{u=1}^{n} (W_U)_i \cdot (1 - T(\widetilde{R}(u^\dagger, u_i), \hat{A}(u_i))) \\
&= \sum_{u=1}^{n} (W_U)_i \cdot I(\widetilde{R}(u^\dagger, u_i), N(\hat{A}(u_i))) \\
&= \sum_{u=1}^{n} (W_U)_{n-i+1} \cdot I(\widetilde{R}(u^\dagger, u_{n-i+1}), N(\hat{A}(u_{n-i+1}))) \\
&= \sum_{u=1}^{n} (W_L)_i \cdot I(\widetilde{R}(u^\dagger, u_{n-i+1}), N(\hat{A}(u_{n-i+1}))) = co\overline{\hat{A}}^{W_L}(u^\dagger).
\end{aligned}$$

The third equality holds form property (2.6h) and the fact that N is the standard negator. In the fourth equality, we replaced indices i with indices $n - i + 1$ and applied the complementarity of W_U and W_L. □

Proposition 8.2.5 states that if $\mathbb{A}$ is N-invariant for N the standard negator, the duality analogous to Eq. (8.5) holds. An example of such an averaging operator is the arithmetic mean.

8.3 Experiments

In this section, we evaluate the performance of FGAC from various perspectives. First, we evaluate the behaviour of granular approximations and FGAC on artificially generated simple datasets to get an empirical impression of how the granular approximations and FGAC handle inconsistencies. In the next step, we compare the prediction performance of different versions of the FGAC as well as OWA-based FGAC using real

data from Table 8.1, together with the encoding of nominal attributes explained in Subsection 8.1.3. In the last step, we compare the prediction performance of the best version of FGAC with the other locally interpretable ML methods from Section 8.1.2.

We implemented the granular approximations, FGAC and all the experiments in the Python programming language [124]. To calculate granular approximations, we use the aforementioned symmetric loss functions: absolute error loss (AEL) (2.11) and squared error loss (SEL) (2.10). The symmetry of the loss functions is important, since we calculate multi-class granular approximations for the purpose of the multi-class classification. In the current version, we used the Łukasiewicz t-norm and the corresponding IMTL triplet in order to evaluate the estimated membership degree (8.4). To solve optimization problems (7.11) and (7.12), we use the Mosek solver [9] and its API for Python. The code for the experiments is available at: `https://github.com/markopalangetic/FGAC_experiments`.

For every model, we select one hyperparameter which will be tuned. Hence, the hyperparameter value for which the model performs best will be chosen. The interpretation of these hyperparameter values is that they control the bias-variance trade-off, i.e., their tuning is used to balance between overfitting and underfitting.

For FGAC, γ will be the hyperparameter that is tuned. We provide an example to illustrate that γ is indeed a parameter that balances between bias and variance.

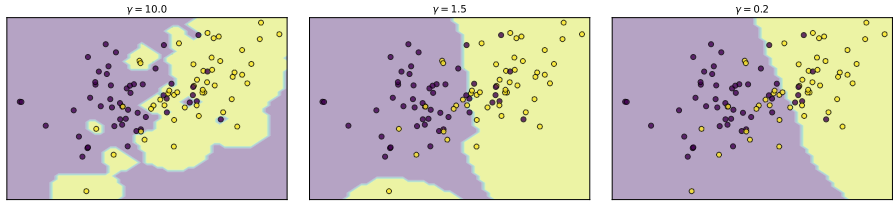

Figure 8.1: Illustrations of decision spaces for different γ

In Figure 8.1, we generated 100 synthetic data instances for a binary classification problem to illustrate the decision areas for different values of parameter γ. The dataset was generated using the SCIKIT-LEARN package and the "make_classification" function. The control of the random number generator is achieved with the command "rand_state=10".

In the left hand image in Figure 8.1, we can see clear overfitting for $\gamma = 10$, as an example of a high value, where the learning process is

affected by the noise in data. As γ decreases, we can see that the decision boundary (the line that separates the two decision classes) becomes smoother and simpler (middle image and right hand image in Figure 8.1) which indicates a less noise-affected learning process. For a very small γ parameter (right hand image), we observe an even simpler decision line which may be a sign of underfitting and indicates that the model did not properly capture the relationship between the condition attributes and the decision attribute.

8.3.1 Simulation study on FGAC and the granular approximations

Before we compare FGAC with other ML methods on real data, we want to observe how it operates and how granular approximations emerge in a more controllable environment. In particular, we create a binary classification problem with artificially generated data, and control the level of inconsistency to observe how FGAC behaves for different levels of inconsistency. In the case of fuzzy similarity (T-equivalence), the level of inconsistency is determined by the non-separability of data instances. The general separability is not formally defined, but we may say that data instances in a classification problem are separable if there exists a simple manifold that separates instances of different decision classes. If there is no such manifold, then the instances from different classes are mixed in the area of the decision boundary, which makes their classification more challenging. If instances from different classes are mixed in the area of the decision boundary, the amount of inconsistency will be larger, since there will be many instances from different classes that are close, and therefore highly similar w.r.t. the given T-equivalence.

In this example, the data is generated in the following way. The instances of two decision classes are generated from two multivariate normal distributions with means $(0,\ldots,0)$ and $(1,\ldots,1)$ respectively. The size of the mean vectors is equal to the number of attributes we have (dimensionality). We ran the experiment for different numbers of attributes, i.e., different dimensions.

The covariance matrix is in both cases the identity matrix multiplied by a constant variance. This variance, or more precisely the standard deviation (the square root of the variance) will be manipulated in order to increase the inconsistency among instances. Namely, when the standard deviation is higher, the instances of two classes are more dispersed, which leads to a situation where the instances of the two classes will in-

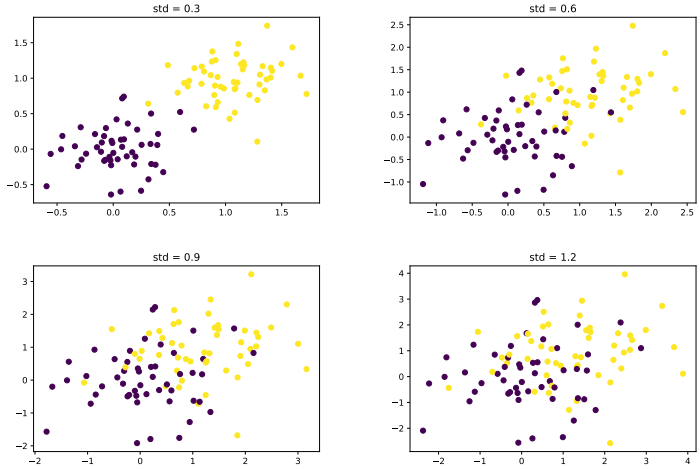

Figure 8.2: Position of instances when the standard deviation is changed

teract more, i.e., they are more mixed in the area of the decision boundary. We see an example in Figure 8.2, where data instances are generated for different values of the standard deviation, which is indicated in the title of every subfigure. We observe that when the standard deviation is 0.3, the instances are almost separable, while increasing the standard deviation leads to instances from different decision classes getting more mixed in the area of the decision boundary, i.e., we expect to observe more inconsistency. We run the experiments with a fixed number of instances which is 1000, i.e., 500 instances per class. We generate data for standard deviations from the set of values $\{0.3, 0.6, 0.9, 1.2\}$ and for numbers of attributes from the set of values $\{3, 10, 30, 100\}$. Therefore, we will consider 16 different combinations. For every such combination, we generate data instances 20 times in order to provide more credibility to the results. The granular approximations are calculated for Euclidean and Chebyshev similarities and for AEL and SEL as loss functions. For each generated dataset, we split it into train data and test data, where the test data compose 20% of the generated data. On the train data, we tune the parameter γ that appears in the T-equivalences. This is obtained using grid search and 5-fold cross validation. After preliminary tests, we decided to tune γ from the following 11 possible values: $\{1/5, 1/3, 1/2, 2/3, 1, 1,25, 1.6, 2, 3, 5, 10\}$. After the optimal γ is obtained, the granular approximation is calculated on the complete train data using the optimal γ. Then, FGAC is applied to the test data, i.e., the mem-

bership degrees are estimated using Eq. (8.7) and compared with the actual decision labels. The evaluation metric on the test data is the one that was used for the training, e.g., if the AEL is used for training, then the performance on the test data will be evaluated using AEL.

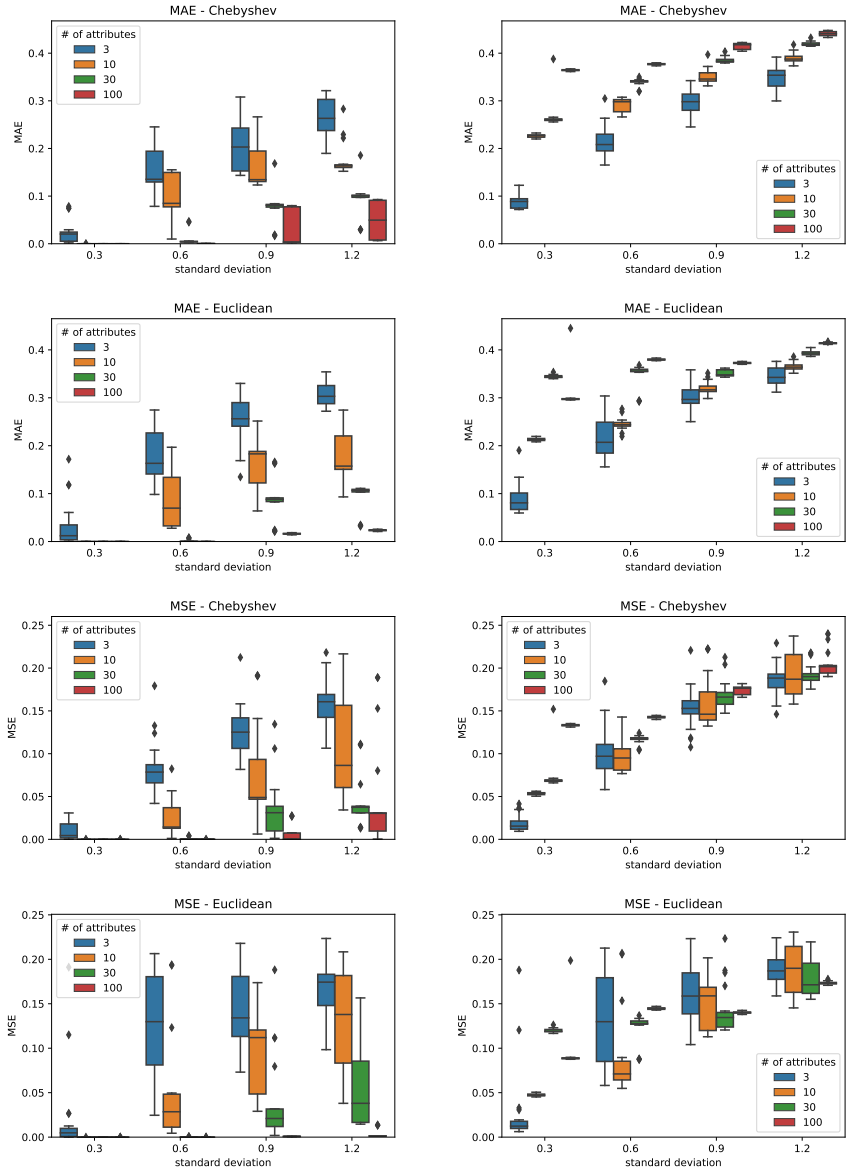

Figure 8.3: Simulation results

The results are shown in Figure 8.3. We have 8 images organized in a 4×2 grid. In every row, a different combination of loss functions (AEL or SEL) and fuzzy similarity (Euclidean or Chebyshev) is used. The combination is indicated in the title of each image. The left hand column of images represents scores on the training data. Namely, after the training is performed on the train data, the score (AEL or SEL) is calculated on both training and test instances. The score on training instances represents the difference between granular approximations and the original fuzzy set (which is crisp in this case), i.e., the optimal risk calculated in Eqs. (6.2) and (6.4). With these scores, we can observe the smallest possible difference between the original labels and a granularly representable set w.r.t. the given loss function. The score on the test instances shows how different the predicted membership degrees are compared with the actual crisp decisions. On all images, we see that the risk is larger for larger standard deviation, which is in line with the reasoning related to Figure 8.2. The next thing we observe is that the training risk is smaller for a larger number of attributes. This is also expected from the geometrical perspective. The generated instances are with a high probability positioned inside a ball of radius equal to 3 times the standard deviation (a well-known statistical rule, [73]). When the dimensionality is increased, the intersection of the balls with centers in $(0,\dots,0)$ and $(1,\dots,1)$ and radius equal to 3 times the standard deviation will be smaller relative to the volume of balls. Therefore, the number of instances from different decision classes that are mixed in the area of the decision boundary will be smaller, i.e., the amount of inconsistency is smaller. This further implies that the calculated training risk will be smaller. However, we observe the general increase of risk on the test data when the number of attributes is larger. This means that FGAC can suffer from the curse of dimensionality, i.e., for higher dimensions we require more data to generalize properly or, in other words, FGAC tends to overfit in high dimensions. The last thing we observe is the smaller variance on the test data. While there are fluctuations in the training risk for different datasets (the size of a single box plot), the test risk is more consistent. One possible conclusion from this is that FGAC as a method is more stable than the calculation of granular approximations.

8.3.2 Comparison on the real data - setup

Next, we compare different versions of FGAC between themselves, as well as with the other locally interpretable ML model.

As we mentioned before, one hyperparameter will be tuned for every model. For the kNN approach, the obvious choice is the number of neighbours k. For the decision tree classifier, we use the maximum depth of the decision tree, while for the LVQ, we choose the number of prototypes trained per decision class.

For the kFRNN models, we are faced with two options. One option is a parameter which controls the number of non-zero instances in OWA vectors. The approach is motivated by [111]. The other approach is the parameter γ from the fuzzy similarity that is used by both kFRNN and FGAC. We choose the first option for two reasons. The first one is that the developers of the kFRNN never used a parameterized version of the fuzzy similarity relation (it was not necessarily a T-equivalence in that case). The second one, which is more important, is that using the γ parameter, while applying OWA-weights on all instances (without zero weights), will lead to a non-interpretable version of kFRNN which makes it incomparable with FGAC in this setting.

In the chosen case, only the first k values of W_U (last k values of W_L) are non-zero. In these experiments, the non-zero values will be those introduced in Section 2.4 (additive, exponential, inverse additive).

We observed that the methods that are based on empirical risk minimization, like FGAC and LVQ are more sensitive to class imbalance that exists in some data sets. Class imbalance is the situation where there is a significantly larger number of instances that are assigned to one decision class than instances assigned to a different one. The larger class is called the majority class while the smaller is the minority class. Because of the class imbalance, an additional preprocessing is desirable.

The most suitable way to handle this for the FGAC is to add extra weight to summands in the empirical risk that correspond to the minority class or classes. However, that would induce an additional advantage to the methods that are based on the empirical risk minimization compared to the other methods. In order to ensure the fairness in the comparison of the ML models, we use the random oversampling method that can be applied to all of them. With random oversampling, we randomly sample instances from the minority classes and add copies of them to the dataset until all decision classes from the training set have an equal amount of instances that is equal to the size of the majority decision class. However, random oversampling may negatively affect some methods and therefore it will not be applied by default; we will keep it as an additional hyperparameter, i.e., the method will decide by itself during the tuning phase if it will apply the random oversampling or not. To

implement the random oversampling, the Python package Imblearn is used.

8.3.3 Comparison of the different versions of FGAC

In order to evaluate the performance of each model, 5-fold cross-validation is used, i.e., the models are trained on 4 folds and evaluated on the fifth one. During each of 5 cross-validation phases, parameter γ is tuned on 4 folds dedicated to training using another, internal 5-fold cross validation. That means that those 4 folds are merged and then split into 5 new folds in order to tune the parameter. This is usually called *nested cross-validation*. The performance evaluation metric used is the balanced accuracy. Parameter γ will be tuned using the same values as before: $\{1/5, 1/3, 1/2, 2/3, 1, 1,25, 1.6, 2, 3, 5, 10\}$. The fine-tuning and cross-validation are implemented in the Scikit-Learn package. The initial seed for the random number generator is set with "rand_state=10" in every situation where required. The names of the columns in the table are composed from the type of the loss function used ("ael" or "sel"), the type of the similarity relation ("Chebyshev" or "Euclidean").

We test if their performance is significantly different from each other. For that purpose, we use the non-parametric Friedman chi-squared test [50]. The null hypothesis of this test is that the performances of the models is indifferent. After running the test, we get that the p-value is of order 10^{-8}, which means that we strongly reject the null hypothesis, i.e., the models are significantly different. The next step is to recognize the best model and to test if it is significantly better than the others. If we look at the average rankings of the models, we have the following:

	ael_Chebyshev	ael_Euclidean	sel_Chebyshev	sel_Euclidean
average rank	3.63	2.109	2.804	1.457

We observe that the model which uses squared error loss and Euclidean similarity has the best average ranking. We hypothesize that this model is the best performing one and we test if this is statistically significant. We use Holm post-hoc analysis [72] as well as its adaptation for comparing machine learning models from [38]. Following the criticism of [38] expressed in [15], we use the Wilcoxon test for the pairwise comparisons. After the Holm procedure is applied, the obtained final p-value is 0.007, which means that we can confidently claim that the best ranked model is significantly the best performing one. The final p-value in this case is obtained as a maximum of the adjusted p-values calculated

during the Holm procedure. We will use the best performing model as a representative version of the FGAC in the comparison with the other ML models.

The next step is to test how the use of OWA operators affects the performance of FGAC. In Table 8.3, we list the results of OWA-based FGAC when min and max are replaced with OWA operators with weights from Section 2.4. As before, the results are given for both SEL and AEL loss functions, as well as for both Chebyshev and Euclidean similarities. First, we test if the 4 models for fixed OWA weights perform differently from each other using the Friedman test. We obtain the following results:

weight:	add	exp	invadd
p-value:	$1.74 \cdot 10^{-5}$	$4.8 \cdot 10^{-6}$	$1.42 \cdot 10^{-6}$

All p-values are very close to 0 which means that the performances are indeed significantly different. If we calculate the average rankings, we find:

	ael_Chebyshev	ael_Euclidean	sel_Chebyshev	sel_Euclidean
add	3.217	2.022	3.087	1.674
exp	3.457	2.196	2.783	1.565
invadd	3.609	2.087	2.652	1.652

As in the non-OWA version of the FGAC, we observe that the best average ranking is achieved for the SEL loss function and the Euclidean similarity. As before, using post-hoc analysis we test if the performance of the best ranked model is significantly better than others. We obtain the following p-values:

	add	exp	invadd
p-value:	0.112	0.035	0.012

From these p-values, we can conclude that for the exponential and inverse additive weights, we can confidently say that the best ranking model performs better than the other models. For the additive weights, the p-value is slightly higher than the usual significance level (0.05). In any case, we will use the best ranking models as representatives of the particular OWA-weights in further comparisons.

In the next step, we compare the performances of the chosen models for different OWA weights with the chosen FGAC model from before. To

recall, we have 4 different models, 3 with different OWA weights (add, exp, invadd) where all 4 models use the SEL loss and Euclidean similarity. After performing the Friedman test on their performances, we get a p-value equal to 0.396 which can be considered high. In other words, we confidently claim that we do not have enough evidence to conclude that using OWA-operators instead of extrema operators will lead to different results. The reason for that may lie in the fact that the learning process is based on the constraints that use extrema instead of OWA operators. The latter are only used in the prediction phase and not in the learning phase. In other words, the learning phase is the key part and adding the OWA operators during the prediction phase cannot improve the results.

For this reason, we exclude OWA-based FGAC from the further analysis.

name	fgac_ael_triangular	fgac_ael_quadratic	fgac_sel_triangular	fgac_sel_quadratic
australian	0.862	0.836	0.868	0.848
breast	0.514	0.634	0.520	0.653
crx	0.729	0.775	0.767	0.778
german	0.506	0.665	0.520	0.666
saheart	0.649	0.687	0.652	0.675
ionosphere	0.931	0.942	0.929	0.937
mammographic	0.792	0.802	0.794	0.801
pima	0.700	0.717	0.709	0.727
wisconsin	0.955	0.973	0.962	0.972
vowel	0.957	0.964	0.961	0.975
wdbc	0.902	0.926	0.900	0.928
balance	0.640	0.678	0.629	0.718
glass	0.596	0.630	0.638	0.639
cleveland	0.319	0.313	0.324	0.315
bupa	0.550	0.564	0.598	0.603
haberman	0.589	0.644	0.613	0.610
heart	0.768	0.807	0.783	0.815
spectfheart	0.585	0.635	0.735	0.741
dermatology	0.918	0.950	0.935	0.950
ecoli	0.665	0.683	0.685	0.716
tictactoe	0.500	0.726	0.500	0.884
vehicle	0.676	0.695	0.685	0.696
sonar	0.771	0.804	0.764	0.876

Table 8.2: FGAC results

name	fgac_ael Chebyshev_add	fgac_ael Euclidean_add	fgac_sel Chebyshev_add	fgac_sel Euclidean_add	fgac_ael Chebyshev_exp	fgac_ael Euclidean_exp	fgac_sel Chebyshev_exp	fgac_sel Euclidean_exp	fgac_ael Chebyshev_invadd	fgac_ael Euclidean_invadd	fgac_sel Chebyshev_invadd	fgac_sel Euclidean_invadd
australian	0.854	0.856	0.853	0.840	0.856	0.846	0.864	0.833	0.858	0.850	0.870	0.845
breast	0.514	0.571	0.516	0.625	0.526	0.645	0.520	0.655	0.514	0.627	0.520	0.648
crx	0.729	0.773	0.728	0.819	0.729	0.758	0.764	0.777	0.729	0.767	0.748	0.779
german	0.504	0.695	0.555	0.650	0.504	0.683	0.525	0.667	0.504	0.689	0.534	0.681
saheart	0.654	0.672	0.668	0.675	0.646	0.674	0.667	0.686	0.660	0.677	0.664	0.684
ionosphere	0.925	0.940	0.929	0.942	0.931	0.940	0.929	0.937	0.931	0.940	0.929	0.937
mammographic	0.792	0.803	0.788	0.816	0.793	0.805	0.791	0.808	0.789	0.805	0.793	0.810
pima	0.710	0.715	0.702	0.699	0.705	0.716	0.709	0.734	0.711	0.715	0.699	0.708
wisconsin	0.942	0.966	0.950	0.962	0.959	0.972	0.956	0.969	0.953	0.970	0.954	0.963
vowel	0.952	0.953	0.954	0.966	0.957	0.960	0.965	0.972	0.956	0.960	0.957	0.971
wdbc	0.893	0.909	0.889	0.909	0.903	0.926	0.904	0.926	0.897	0.920	0.898	0.919
balance	0.800	0.724	0.800	0.759	0.742	0.717	0.719	0.649	0.733	0.700	0.746	0.684
glass	0.570	0.575	0.549	0.574	0.587	0.635	0.624	0.624	0.562	0.584	0.604	0.660
cleveland	0.334	0.394	0.306	0.348	0.313	0.333	0.318	0.325	0.325	0.337	0.337	0.356
bupa	0.558	0.585	0.532	0.552	0.580	0.577	0.613	0.589	0.571	0.582	0.546	0.586
haberman	0.619	0.612	0.597	0.622	0.611	0.591	0.614	0.626	0.583	0.613	0.623	0.609
heart	0.776	0.825	0.784	0.829	0.762	0.810	0.776	0.837	0.765	0.815	0.774	0.838
spectfheart	0.523	0.518	0.637	0.616	0.569	0.628	0.720	0.732	0.546	0.546	0.687	0.691
dermatology	0.917	0.964	0.932	0.967	0.932	0.950	0.943	0.954	0.940	0.950	0.944	0.955
ecoli	0.653	0.692	0.698	0.712	0.668	0.684	0.712	0.714	0.634	0.685	0.709	0.693
tictactoe	0.500	0.801	0.500	0.956	0.500	0.773	0.500	0.895	0.500	0.797	0.500	0.944
vehicle	0.666	0.676	0.682	0.692	0.675	0.695	0.678	0.701	0.672	0.689	0.697	0.706
sonar	0.730	0.740	0.713	0.810	0.766	0.783	0.762	0.881	0.738	0.773	0.747	0.866

Table 8.3: FGAC results for different OWA weights

8.3.4 Comparison of FGAC with other ML methods

We first discuss how the hyperparameters are tuned. We already stated previously that every model depends on one parameter and we tune that parameter using 5-fold cross-validation. They are selected from a finite set of values based on their performance. In the following table, we list the models and the corresponding sets of possible values of their hyperparameters.

models	possible hyperparameter values
FGAC	{1/5, 1/3, 1/2, 2/3, 1, 1,25, 1.6, 2, 3, 5, 10 }
kFRNN	{ all, 1, 3, 5, 10, 15, 20, 25, 30, 40, 50}
kNN	{1, 3, 5, 7, 10, 15, 20, 25, 30, 40, 50}
LVQ	{1,2,3,4,5,6,7,8, 9,10, 11}
CART	{2,3,4,5,6,7,8, 9,10, 11,12}

The possible values are constructed based on the preliminary analysis. Every model is provided with 11 possible hyperparameters. Value "all" in the kFRNN hyperparameters set indicates that the OWA weights were applied to all instances. Also, after preliminary analysis, we concluded that the best performing version of kFRNN is the one with OWA additive weights and that uses Euclidean similarity and hence, it is used in the comparison process as the representative of kFRNN.

In Table 8.4, we show the performances of the models. In every row, with the black bold font, we label the best performing model. After running the Friedman test on the results, we obtain a p-value equal to 0.0235, which implies that there is some evidence to claim that the models are significantly different.

In the next table, we show the average rankings of these models.

models	FGAC	kFRNN	kNN	LVQ	CART
average rank	2.565	3.696	2.37	2.978	3.391

First, we observe that FGAC has the second best performance based on the average rank; the only better model is kNN. However, if we apply the post-hoc test to check if kNN is indeed better than all the other methods, the obtained p-value is 0.3, i.e., we do not have evidence to claim that.

name	FGAC	kFRNN	kNN	LVQ	CART
australian	0.848	0.816	**0.867**	0.834	0.853
breast	**0.653**	0.639	0.632	0.644	0.637
crx	0.778	0.732	0.835	0.760	**0.861**
german	0.666	0.612	0.683	**0.684**	0.648
saheart	0.675	0.672	**0.680**	0.674	0.624
ionosphere	**0.937**	0.850	0.830	0.853	0.879
mammographic	0.801	0.800	0.817	0.802	**0.829**
pima	0.727	0.711	**0.745**	0.719	0.702
wisconsin	**0.972**	0.963	0.970	0.970	0.953
vowel	0.975	0.977	**0.986**	0.809	0.785
wdbc	0.928	0.936	**0.956**	0.937	0.929
balance	0.718	**0.793**	0.717	0.615	0.596
glass	0.639	0.609	0.641	0.572	**0.654**
cleveland	0.315	0.308	**0.339**	0.328	0.254
bupa	0.603	0.611	**0.628**	0.602	0.623
haberman	0.610	0.593	0.595	0.632	**0.643**
heart	0.815	0.778	0.829	**0.841**	0.754
spectfheart	**0.741**	0.694	0.737	0.662	0.642
dermatology	0.950	0.934	0.944	0.953	**0.959**
ecoli	0.716	**0.743**	0.677	0.712	0.598
tictactoe	0.884	0.900	0.877	**0.973**	0.930
vehicle	0.696	0.681	**0.710**	0.622	0.695
sonar	**0.876**	0.757	0.862	0.825	0.709

Table 8.4: Comparison of the FGAC with the other ML models based on the balanced accuracy

	kNN	kFRNN	LVQ	CART
p-values:	0.3	0.016	0.134	0.065

Table 8.5: Pairwise comparison of the FGAC with other models

We check if FGAC is significantly different from the other methods. If we apply the Wilcoxon test to make pairwise comparisons of FGAC with the remaining models, we obtain the p-values in Table 8.5. We observe that FGAC is only significantly better than kFRNN. If we run the post-hoc analysis to check if FGAC is better than all the others models except the kNN, the obtained p-value is 0.134 which means that we can-

not support such a claim.

In the next section, we discuss the greatest advantage of the FGAC - its interpretability.

8.4 Interpretability

In this section, we discuss the interpretability of the proposed FGAC method and we compare it with the interpretability of the other methods. We first discuss the method from the fuzzy logic perspective, i.e., its possibility to be translated into linguistic expressions. These linguistic expressions can provide insight about how the model works as a whole, but since they do not interpret the parameters of the model, we cannot consider this as a modular interpretability.

The second part of the section is related to the local interpretability of FGAC. We try to identify arguments, which are instances from the training set, that are "in favour" or "against" the estimated membership degree of a new instance. In this way, we show that sometimes other instances can be used to explain a particular prediction and we discuss when this is suitable. At the end, we compare the local interpretability of FGAC with the ML methods from Section 8.1.2.

8.4.1 Fuzzy logic and linguistics

The goal of this section is to interpret the expression (8.1) and its multi-class version, i.e., we explain these inequalities by utilizing the ability to express the fuzzy connectives using plain words. We interpret a T-equivalence relation as "similarity", t-norms as the "and" connective and implicators as IF-THEN rules.

First, we interpret the well-definedness of the bounds expressed through Proposition 8.2.1, as well as the proof of the proposition.

An equivalent form of the well-definedness of the bounds is given in Eq. (8.2). For some $u, v \in U$, the interpretation of that expression is:

$$\text{IF } u \sim u^{\dagger} \text{ and } u \widetilde{\in} A \text{ THEN IF } v \sim u^{\dagger} \text{ THEN } v \widetilde{\in} A, \tag{8.10}$$

where $\sim$ means "is similar to" and $\widetilde{\in}$ stands for fuzzy membership, i.e., we read it as "belongs to". Therefore, we read the previous expression as "If u is similar to $u^{\dagger}$ and u belongs to A then, if v is similar to $u^{\dagger}$ then v is in A".

Following the proof of the proposition, the previous expression is equivalent to (residuation property):

$$\text{IF } u \sim u^{\dagger} \text{ and } v \sim u^{\dagger} \text{ and } u \widetilde{\in} A \text{ THEN } v \widetilde{\in} A,$$

which is true from the T-transitivity of $\sim$ and the granularity property. Since expression (8.10) holds for all u and v, it can be translated to:

$$\text{IF } \exists u \in U \text{ s.t. } u \sim u^{\dagger} \text{ and } u \widetilde{\in} A \text{ THEN } \forall v \in U \text{ IF } v \sim u^{\dagger} \text{ THEN } v \widetilde{\in} A.$$

Here, the symbols $\exists$ and $\forall$ have their usual meanings: "there exists" and "for all" respectively, while "s.t." is an abbreviation for "such that". Putting back the membership degree of $u^{\dagger}$, the two inequalities of (8.1) can be interpreted as follows. For the left inequality we have:

$$\text{IF } \exists u \in U \text{ s.t. } u \sim u^{\dagger} \text{ and } u \widetilde{\in} A, \text{ THEN } u^{\dagger} \widetilde{\in} A, \tag{8.11}$$

while for the right inequality, we have that:

$$\text{IF } u^{\dagger} \widetilde{\in} A, \text{ THEN } \forall v \in U, \text{ IF } v \sim u^{\dagger} \text{ THEN } v \widetilde{\in} A. \tag{8.12}$$

We apply the previous expressions on our example with the movie streaming service. From (8.11) we have that: if there exists a movie u that is similar to movie $u^{\dagger}$ and the user likes movie u, then the user will also like movie $u^{\dagger}$. From (8.12) we have that: if the user likes movie $u^{\dagger}$ then they should also like all movies that are similar to u.

8.4.2 Instance-based interpretability

Getting the arguments

The next step is to identify and to interpret the training instances based on which the decision for a new instance was made. These instances are argmax from the left equation and argmin from the right equation in (8.1). The argmax is the instance that supports the decision $u^{\dagger} \in A$, since it is at the same time the most similar to $u^{\dagger}$ and has the highest estimated membership in A. All other instances are either less similar to $u^{\dagger}$, or less present in A. Hence, the argmax is the argument in favour of decision $u^{\dagger} \in A$. The argmin is the instance that objects the decision $u^{\dagger} \in A$, since it supports the decision $u^{\dagger} \in coA$. This is visible by applying negator N to the right inequality of (8.1) and obtaining $N(\hat{A}(u^{\dagger})) \geq \max_{u \in U} T(\widetilde{R}(u, u^{\dagger}), N(\hat{A}(u)))$. After obtaining the previous expression, we can use the reasoning from above to justify that the

argmin indeed supports $u^\dagger \in coA$, i.e., objects $u^\dagger \in A$. In other words, the argmin is the argument against the decision $u^\dagger \in A$.

The conclusion of the previous paragraph is that we are able to find arguments in favour of the decision, as well as arguments against the particular decision. If we need more than one argument for the decision, we can consider a few top instances (not only minimum and maximum) that support and that object the decision. In our example of movie recommendations, for every movie for which we predict the degree of allure to the user, we can identify the movies that support this degree and the movies that object the degree from the movies that the user already watched and rated. Moreover, for arguments that are in favour of a decision, value $T(\widetilde{R}(u^\dagger, u), \hat{A}(u))$ can be seen as the strength of the argument. The greater the strength, the more confident we are about our decision. On the other hand, for arguments that go against the decision, value $T(\widetilde{R}(u, u^\dagger), N(\hat{A}(u)))$ can be seen as the strength of the argument. If the value is greater, then value $I(\widetilde{R}(u, u^\dagger), \hat{A}(u))$ is smaller which further implies that the confidence in our decision is also smaller.

Since we are able to precisely identify the arguments based on which the decision was made and since those arguments can be well comprehended by a human, we may say that FGAC is fully locally interpretable.

Didactic example

Here we demonstrate how classification arguments are identified in practice. Suppose we have a task to identify hate speech from text. The dataset consists of short texts that were found on social networks together with labels indicating if a particular text is considered hate speech or not. The labeling of text is done manually and therefore it depends on an individual's interpretation of hate speech and personal political beliefs. The dataset was a part of the Semeval-2019 competition and it was downloaded from the official website of the competition [13]. The complete dataset consists of 13 000 instances divided into 3 groups: train, development and test data of the corresponding sizes 9000, 1000 and 3000. However, for didactic purposes, we use only the development set with 1000 instances. We also note that hate speech detection may be seen in a gradual manner and that different texts may possess different amounts of hate speech, which makes it a suitable task for using fuzzy membership degrees.

The first step is to perform an embedding of text into a high dimensional Euclidean space such that the cosine similarity of the instances

corresponds to the semantic similarity of the text fragments, i.e., to determine whether the fragments are talking about a similar topic or not. For this purpose, we use the language model RoBERTa and its version that is specialized for hate speech [92, 10]. Using the RoBERTa model, we assign to each text fragment a numerical vector of size 768 that represents its embedding. At the end, we have 1000 instances with 768 attributes that are labelled with 1 (hate speech) or 0 (no hate speech). Therefore, we deal with a binary classification problem where we will apply FGAC with the similarity relation based on the inner product (3.8).

The data is first split into train and test sets where the test set possesses 20% of all data. On the train data, we perform grid search using 5-fold cross validation in order to tune the parameter from Eq. (3.8). After the model is trained and the parameter is tuned, the model is evaluated on the test set and the obtained balanced accuracy score is 0.786. We now provide two examples how particular predictions are obtained. In the first example, we have the following text fragment from the test set.

We had plenty of diversity before the #Globalist
elites started to import the 3rd world.
#StopMassMigration #BuildTheWall #DeportThemAll
#DeportIllegalAilens #NoAmnesty #NoDACA #BuildTheWall

Figure 8.4: Example - first text fragment

The text from Figure 8.4 evidently possesses a negative sentiment on the illegal immigration issue in the United States of America (USA). It was considered as hate speech during the labeling process and it was predicted as hate speech by FGAC.

In Table 8.6, we can see the 3 text fragments from the training set that are considered as the strongest arguments in favour of classifying the fragment from Figure 8.4 as hate speech. Besides that, in the column "similarity", we have the evaluation of the T-equivalence between the text from Figure 8.4 and the fragment from the corresponding row. In column "degree of hate speech", we show the estimated level of hate speech present in the corresponding fragment, while column "argument strength" contains the values obtained using the aggregation with T_L of the values from the columns "similarity" and "degree of hate speech". These values are the actual strengths of the arguments, i.e., they show how much the fragments from the corresponding rows "drag" the text

from Figure 8.4 into class "hate speech". We see that all the arguments exhibit a negative view towards the immigration in the USA and they support building a wall at the border with Mexico, in the same manner as the text fragment from Figure 8.4.

In Table 8.7, we show the 3 text fragments from the training set that are considered as the strongest arguments against of classifying the text from Figure 8.4 as hate speech. The description of the table is the same as for Table 8.6, were the only difference is that now we have "degree of non-hate speech", which is the membership degree to the opposite class. It is obtained as the fuzzy negation of the degree of hate speech. We now see that all the arguments have a very low strength (close to 0). The strongest argument is the one that is still relatively similar to the text from Figure 8.4 (they both call to "build the wall"), but the argument does not have a high membership in the non-hate speech class and therefore the total strength is small. This leads to the conclusion that no instances strongly oppose to classifying the text from Figure 8.4 as hate speech. The resulting degree of hate speech is obtained as the average of the largest strength in favour and the negation of the largest strength against, i.e., $\frac{0.612+1-0.024}{2} = 0.794$. This expression is in accordance with Eqs. (8.4) and (8.6).

	text	similarity	degree of hate speech	argument strength
1	DACA-age illegals far more likely to commit crimes,be in jail @POTUS@WhiteHouse@HouseGOP @SenateGOP #EndDACA #NoAmnesty #EnforceUSLaws8USC1324-25#EVerify#EndChainMigration #EndVisaLottery #BuildTheWall Immigration reform starts with clean slate	0.688424	0.923594	0.612018
2	@realDonaldTrump #MyBad #StopTheInvasion #GreenCardsForDACA #NewChainMigration No-Lottery #IllegalSentHome get in Line #BuildTheWall	0.682003	0.925965	0.607968
3	@FoxNews We have enough of our own Monsters why do we continue to import more Monsters from other Countries? #BuildTheWall, #SecureTheBorder, . #EnforceImmigrationLaws, #EndChainMigration, #MakeEVerifyMandatory	0.611523	0.950577	0.562100

Table 8.6: Top 3 arguments in favour of labeling the text from Figure 8.4 as hate speech

	text	similarity	degree of non-hate speech	argument strength
1	Texas woman, 21, dies after falling from moving SUV, may have been pushed by illegal alien driverhttps://t.co/3t0PH9Hd0s #InOurBackyard #BuildTheWall #PreventableDeath #SecureTheBorder #StopTheInvasion #TherIsMoreOfThemOutThere #AllIllegalAliensAreLawbreakers	0.570377	0.453507	0.023884
2	@RealDonaldTrump Lowest Black Unemployment in History! Enforcing #Immigration laws means Illegals are no longer taking jobs, lowering American Wages and destorying Black ommunities likeCompton CA No #DACA #WalkAway #BlackTwitter #BlacksForTrump @RealC	0.467787	0.535955	0.003743
3	@MSNBC If the refugees dont get food and water - they will go back quick, here the President havnt to do much !	0.000000	0.260544	0.000000

Table 8.7: Top 3 arguments against labeling the text from Figure 8.4 as hate speech

Next, we present an example where FGAC classified the given text fragment differently than the original labeling. Consider the text displayed in Figure 8.5. At first glance, the text looks as a small report (or as the beginning of a report) about the illegal immigration and not as a hate speech as such. However, in the training set it is labeled as hate speech.

> About 25% of illegal crossers have a criminal record in the US and an unknown percentage have committed crimes in their home countries. Of the 92 migrants, 65 had no criminal records. Ten were parents, all...

Figure 8.5: Example - second text

In Table 8.8, we can find arguments in favour of FGAC's decision, while in Table 8.9, the arguments appear against classifying the text from Figure 8.5 as hate speech. Among the arguments in favour, we see some fragments considered as hate speech with a high degree, but with a low similarity to the text from Figure 8.5, leading to a weak strength of the best argument (0.247). The arguments against are fragments that are also reports on illegal immigration and are considered to display a very low level of hate speech. As can be noticed, they are also not considered as similar to the text from 8.5. However, the strongest argument possesses a sufficiently high similarity of 0.455 in order to boost its argument strength to 0.296. This strength is larger than the highest strength of arguments in favour (0.247) which will lead to an estimated degree of non-hate speech of $\frac{0.296+1-0.247}{2} = 0.5245 > 0.5$, i.e., the text fragment from Figure 8.5 will be classified as non-hate speech.

	text	similarity	degree of hate speech	argument strength
1	US immigrants 'living in fear' of Trump's deportation drive @AJENews https://t.co/HROiJwCczUThere's no need to live in fear. All these illegals can pack up and leave. Take your parents, siblings, aunts and uncles, even your friends. Don't live in fear just leave.	0.408164	0.838343	0.246507
2	India should be tough on illegal immigration from Bangladesh and deport the immigrants. Once these people settle down they slowly move south. https://t.co/qcDP8pTC8G	0.377761	0.863047	0.240808
3	#IllegalImmigrants #IllegalAliens #ElectoralSystem #ElectoralCollege I'm going to shock some people here: America is NOT a #Democracy, America is a #Republic. Even more defined America is a Representative Republic. In a TRUE... https://t.co/kcZqVEaR93	0.389923	0.849346	0.239269

Table 8.8: Top 3 arguments in favour of labeling the text from Figure 8.5 as hate speech

	text	similarity	degree of non-hate speech	argument strength
1	As a devastating report reveals 300,000 illegal migrants are living in one French suburb https://t.co/swLCtPvQIC	0.455458	0.840934	0.296392
2	* Sweden: The Afghan migrant whose deportation was thwarted by a naive and “attention-seeking” student activist was actually sentenced for assault and received a prison sentence in Sweden. https://t.co/hccQmt7KMT #v4 #visegrad https://t.co/2Qo8friTwB	0.370020	0.866108	0.236128
3	What is actually happening is very different. According to the U.N.H.C.R the breakdown of refugees are 13% women, 12% children, 75% men aged between 19 to 45. These are not the demographics of people fleeing a war. https://t.co/295OTvmhmv	0.331276	0.895162	0.226439

Table 8.9: Top 3 arguments in favour of labeling the text from Figure 8.4 as hate speech

8.4.3 Interpretability comparison with other models

We discuss the position of FGAC w.r.t. instance-based models and methods discussed in Section 1.4. FGAC is a locally interpretable model, trained directly on data and it is not used in explaining other black-box models in this setting. Therefore, we can compare FGAC with locally interpretable instance-based models directly, and we can compare it with instance-based methods for explaining black-box models from the perspective of the approach they undertake to complete their tasks. We first discuss the latter.

The four approaches for explaining black-box models that were discussed in Section 1.4.3 are counterfactuals, adversarial examples, prototype-based methods and influential instances.

The methods based on counterfactuals from Section 1.4.3 generate new artificial instances as explanations. In this case, one solves an optimization problem for which the solutions are instances from the attribute space that are assigned to the opposite decision class (multiple solutions of the optimization problem are possible). There are two ways to report explanations with counterfactuals: using the generated instances or identifying attributes that have been changed. The generated instances, similarly as prototypes in LVQ discussed in Section 1.4.2, do not have to be meaningful as they are the output of an optimization problem. For example, if pixels of an image are changed, the resulting image does not have to represent any meaningful object. Therefore, reporting such counterfactual does not contribute to the interpretation. A similar conclusion can be derived when reporting the changed attributes; identifying pixels in an image that changed does not necessary lead to any meaningful conclusion if those pixels do not form a recognizable object. From this reasoning, we see that counterfactuals are useful when the vast majority of possible instances from the attribute space are meaningful, which is usually the case for tabular data with numerical attributes. On the other hand, FGAC is independent from the attribute representation of the instances, but it suffers if training instances are not meaningful for the explanations.

The other three instances-based methods discussed in Section 1.4.3 can be considered as globally oriented, i.e., they tend to explain models as a whole. The adversarial examples create meaningful instances in order to deceive the model as a whole. However, as already noted in the introduction, these examples are giving just the basic insights and cannot provide an overall interpretation of the model, which is not their primary aim. The prototype-selection methods also tend to ex-

plain the model globally by identifying the representatives of the data distribution (prototype and critics) from the training dataset. The influential instances, as the last such method, aim to find the instances which removal can severely affect the prediction performance of the whole model. Again, we spot the tendency towards global explanations here. As methods that want to explain black-box models as a whole, they are essentially different from FGAC which is oriented to explain individual predictions.

In the remainder of this subsection, we compare the interpretability of the proposed FGAC method with the ML models from Section 8.1.2. These models are divided into three groups: instance-based (kFRNN and kNN), prototype-based (LVQ) and rule-based (CART). All these types of models possess some form of local interpretability and this is the reason they are selected for this comparison experiment.

In the case of CART, for every performed classification, we are able to identify the corresponding decision rule from the tree structure of the classifier based on which the classification is performed. At the global level, the set of all decision rules, together with the hierarchical structure, can be seen as a form of global interpretability. However, in practice, the number of rules can be very large which aggravates the understanding of the model as a whole. If the number of rules is kept relatively small (e.g. less than 10), we may say that we also achieve global interpretability. On the other side, decision rules depend on the attributes used in the modeling and any feature engineering process may affect the interpretability of CART. On the other side, FGAC is not dependent on the attribute space used for modeling and therefore, it can be advantageous in a context where we have meaningful instances without meaningful attributes (like texts or images). However, the interpretation of rules has its advantages in a way that we are able to exactly identify the way one attribute affects the final decision.

For the LVQ method, we observe that during the training phase, few points in the attribute space are learned as prototypes for every decision class. Later on, the decision is made based on the closest prototype. Prototype-based and instance-based (like FGAC) methods share similarities in a way that both methods make predictions based on the closest points from the attribute space. The difference is that in prototype-based methods, these points are not from the set of training instances, but they can be any points from the space. This is a huge disadvantage if a certain amount of feature engineering is applied and the original attribute space is changed: the learned prototypes lose their meaning and the method

loses interpretability. On the other hand, the interpretability of FGAC does not depend on feature engineering. Therefore, the interpretability of FGAC showes some advantages compared to LVQ.

Now we move to the remaining methods, kNN and kFRNN, which are both instance-based, i.e., of the same type as FGAC. Their possible interpretability lies in identifying instances based on which a prediction was made. Their interpretability heavily depends on the number of instances used for prediction making, i.e., hyperparameter k. If k is high, it is really hard to identify how the prediction is made. We observed that during training of kNN and kFRNN, the majority of performances from Table 8.4 are achieved for higher values of k ($k > 5$) which means that in the majority of cases, the prediction process in both kNN and kFRNN is barely interpretable. Also, kNN and kFRNN are not significantly better than FGAC according to Table 8.5.

Now, we want to compare the FGAC with the more interpretable variants of kNN and kFRNN. We consider a similar interpretability level as for FGAC, i.e., $k = 1$ and a less interpretable case when $k \leq 5$. The comparison results are shown in Table 8.10. Bold values indicate the best performing model. After applying the Friedman test to the results in Table 8.10, we get a p-value of order 10^{-9}, which means that the performances are indeed different. From the table, we observe that FGAC is the best model in most occurrences. Using Holm post-hoc analysis, we test if FGAC is indeed the best model and we get that the p-value is equal to 0.038. This means that FGAC is indeed the best performing model among the selected interpretable instance-based classifiers.

8.5 Conclusion

In this chapter, we introduced the Fuzzy Granular Approximation Classifier (FGAC) based on granular approximations and their multi-class version introduced in Chapter 6 and 7, respectively. We also introduced a version that uses OWA operators. Furthermore, we discussed ways to speed up the training of the classifier. The empirical comparisons led to the following main conclusions:

- The best performing version of FGAC is the one that uses SEL as the loss function and the Euclidean similarity.
- Adding OWA operators does not change the performance of FGAC.
- In comparison with other models, FGAC was the second best

	FGAC	kNN ($k \leq 5$)	kFRNN ($k \leq 5$)	kNN ($k = 1$)	kFRNN ($k = 1$)
australian	**0.848**	0.830	0.808	0.805	0.780
breast	**0.653**	0.591	0.600	0.575	0.568
crx	0.778	**0.829**	0.727	0.784	0.732
german	**0.666**	0.617	0.586	0.631	0.560
saheart	**0.675**	0.621	0.617	0.611	0.576
ionosphere	**0.937**	0.881	0.850	0.829	0.839
mammographic	0.801	**0.804**	0.785	0.744	0.730
pima	**0.727**	0.708	0.664	0.658	0.641
wisconsin	**0.972**	0.966	0.966	0.945	0.938
vowel	0.975	**0.986**	0.976	**0.986**	0.977
wdbc	0.928	**0.957**	0.933	0.948	0.929
balance	**0.718**	0.611	0.623	0.571	0.555
glass	0.639	0.654	0.626	**0.663**	0.615
cleveland	0.315	0.303	**0.320**	0.279	0.291
bupa	0.603	**0.642**	0.614	0.635	0.613
haberman	**0.610**	0.565	0.530	0.536	0.522
heart	**0.815**	0.813	0.758	0.771	0.744
spectfheart	**0.741**	0.702	0.656	0.600	0.634
dermatology	**0.950**	0.940	0.915	0.943	0.904
ecoli	**0.716**	0.680	0.702	0.675	0.675
tictactoe	0.884	0.746	**0.885**	0.750	0.864
vehicle	0.696	**0.719**	0.685	0.688	0.664
sonar	**0.876**	0.862	0.744	0.872	0.725

Table 8.10: Comparison of FGAC with the interpretable versions of kNN and kFRNN

model. However, after pairwise significance testing with other models, we cannot claim that FGAC significantly different than the other models.

Later, we showed that FGAC can be presented using plain words due to the linguistic nature of fuzzy logic. We also classified the method as locally interpretable where for every prediction we are able to identify the arguments for that prediction that are both in favour and against. Finally, we discussed in which cases FGAC is more advantageous compared to other models regarding local interpretability.

Chapter 9

Epilogue

9.1 Conclusion and contributions

In this dissertation, we tackled the problem of inconsistency in data from various perspectives including traditional rough sets and fuzzy rough sets, as well as new approaches based on statistical learning and optimization. Also, we dedicated a significant amount of work to the exploration of the granular properties of the proposed methods. At the end, we explored how the newly developed approaches can be used in classification problems.

For each out of the 5 main chapters, we highlight our main contributions:

- In Chapter 4, we first unified the definition of IRSA and DRSA into PRSA. While the definition of fuzzy IRSA existed previously, we extended it to DRSA as well through the definition of fuzzy PRSA. We proved various important properties and discussed how fuzzy PRSA can be enriched with the OWA approach in order to make it more robust. While OWA-based fuzzy IRSA was investigated in literature, OWA-based fuzzy DRSA is a novelty. We also provided empirical evidence that adding OWA to fuzzy DRSA indeed enhances its robustness.

- In Chapter 5, we discussed the granular properties of fuzzy PRSA and OWA-based fuzzy PRSA. While the granular properties of IRSA, DRSA and fuzzy IRSA were already known, we provided a new view of the granularity from the perspective of the newly introduced PRSA and fuzzy PRSA models. On the other side,

as a completely new result, we discussed the granular properties of OWA-based fuzzy PRSA. In particular, we showed that for D-convex t-norms, the OWA-based fuzzy PRSA approximations are also granularly representable fuzzy sets. At the end of the chapter, we provided a characterization of D-convex t-norms and an example how such fuzzy connectives can be constructed.

- In Chapter 6, we tackled the problem of inconsistency from the statistical learning perspective, motivated by the Kotłowski-Słowiński approach that considered the same problem for crisp relations. We extended their approach for general fuzzy T-preorder relations. The concept of granular approximation was introduced as a generalization of the fuzzy rough approximations; it is obtained as a result of an optimization problem developed using the statistical learning theory. We also showed how to solve such optimization problems in practice and provided didactic examples to illustrate what can be modeled with the approach.

- In Chapter 7, we examined granular properties of the granular approximations. We introduced the concepts of disjoint and adjacent granules which are defined based on the relationships among the granules. These new concepts then helped us to extend the granular approximations to the multi-class classification case, leading to the definition of the multi-class granular approximation. We formulated an optimization problem which enables us to obtain the approximation and we discussed how to solve such problem. At the end, visual examples of the new granular concepts were provided.

- In Chapter 8, we tested how the granular approximations perform in prediction tasks. A Fuzzy Granular Approximation Classifier (FGAC) was introduced as a native extension of the granular approximation for prediction purposes. We compared its classification performance with other similar ML methods and we discussed its biggest strength: transparency, i.e., the ability to clearly explain how a particular prediction was obtained. We showed that while other similarly transparent methods do not perform significantly different than FGAC, the transparency of the latter is superior.

9.2 Future challenges

The results we presented can be expanded in various directions. During our work, we identified several candidate problems where further contributions can be achieved.

9.2.1 Classification

First, we discuss the possibilities to improve the presented classification model FGAC. We list the following options.

- In the experimental evaluation of FGAC, a T-equivalence relation was used that is suitable for ordinary classification problems. On the other side, using a non-symmetric T-preorder relation is more suitable for monotone classification problems. Since the binary version of FGAC is developed also for non-symmetric relations, a direction for future research is to explore its performance in monotone classification problems.

- In the same experiments, we used fuzzy connectives based on the Łukasiewicz t-norm. Another possibility is to explore if using different fuzzy connectives, isomorphic to the Łukasiewicz ones, or in general different fuzzy connectives, can lead to better results.

- In order to obtain granular approximations, the corresponding optimization problems are solved by putting their formulations to an existing optimization solver. Despite the fact that modern-day solvers are very efficient in solving optimization problems, smarter, purpose-built implementations can lead to higher time savings during the training phase of FGAC.

9.2.2 Regression

Another important question is how can granular approximations be used for regression problems. We saw a didactic example of inconsistency in a regression problem in Subsection 3.2.2 as well as how it was handled by granular approximations in Section 6.2. However, the whole application in the regression problems depends on the transformation of the eventual decision values into fuzzy membership degrees (e.g., transformation of prices to a degree of expensiveness in Subsection 3.2.2). We here present a possibility to formulate an inconsistency correction

optimization problem by using a fuzzy relation on the decision attribute instead of the transformation.

Granular approximations rely on the consistency property which can be formulated as:

$$\widetilde{R}(u,v) \leq I(A(u),A(v)),$$

for instances u and v. If $\widetilde{R}$ is symmetric, we can exchange the positions of u and v on the right side and we have the combined condition

$$\widetilde{R}(u,v) \leq \min(I(A(u),A(v)),I(A(v),A(u))) = Eq(A(u),A(v)).$$

Operator Eq on the right side is a fuzzy equivalence relation which measures "how identical" or, in other words, how similar $A(u)$ and $A(v)$ are. This reasoning can further be extended to a general regression problem. Instead of measuring similarity of two fuzzy values, we can measure the similarity of two arbitrary real values that we observe in a regression problem. In other words, let $\bar{y}_u \in \mathbb{R}, u \in U$ be observed values of the decision attribute in a regression problem while $\hat{y}_u \in \mathbb{R}, u \in U$ are the values that should be estimated. Let L be a loss function, $\widetilde{R}_X$ a T-equivalence relation on the condition attributes and $\widetilde{R}_y$ a T-equivalence relation on the decision attribute (that depends on $\hat{y}_u$ and $\hat{y}_v$). Then, the regression problem can be formulated as

$$\begin{aligned} &\text{minimize} && \sum_{u \in U} L(\bar{y}_u, \hat{y}_u) \\ &\text{subject to} && \widetilde{R}_X(u,v) \leq \widetilde{R}_y(u,v), \quad u,v \in U. \end{aligned} \tag{9.1}$$

Such problem requires a lot of restrictions on relation $\widetilde{R}_y$ in order to formulate the constraints as linear or to ensure the convexity of the optimization space.

With (9.1) we formulated the learning procedure of a new regression model. The natural next step would be to develop the prediction phase and to compare its performance to that of other similar ML models. Also, in the same way we formulated the transparency of FGAC, it would be interesting to investigate if similar transparency properties can be inherited for this regression model.

9.2.3 Rule induction

Two chapters of this thesis were dedicated to the granular properties of the obtained granular and fuzzy rough approximations. As already stressed, these granular properties are important since they enable us to

develop prediction models based on rule induction. The interpretation of such rules was recalled in Chapter 5.

Therefore, some of the challenges to tackle include:

- We observed that in the granular representation, every instance generates a rule. That implies that the number of possible rules is equal to the number of instances which is infeasible in practice. One of the challenges is to find a proper rule induction algorithm to reduce the number of covering rules. It would be desirable to explore how such algorithm would perform in classification tasks as well as if it is possible to control the trade-off between transparency and predictive capability.

- Apart from the rule selection procedure discussed in the previous point, one may try to merge granules that correspond to different instances in order to obtain a "supergranule" which can cover a larger number of instances but still be interpreted as a single rule.

- In (fuzzy) rough set theory, the concept of reducts is used in attribute selection procedures and during rule induction to reduce the length of individual rules. It would be worth to explore if the novel granular approximations can be used for similar purposes.

- It is well-known that rule-based methods, like CART, perform very well when combined with ensemble procedures like bagging or boosting [71, 134]. Exploring a similar integration with rule induction methods based on granular approximations would be an interesting proposition.

9.2.4 Interpretability

In Chapter 8, we discussed the interpretablity of the FGAC model. The question arises if FGAC, as an interpretable model, can be used as a model-agnostic approach (global or local) to explain black-box models in the similar manner as linear or rule-based models are used. The first step should be to check on the global interpretability, i.e., to apply FGAC on data which decision labels were changed according to the black-box model. It would be interesting to observe how FGAC would behave in such a situation. The more challenging part is to try to apply FGAC as a local

model-agnostic method, since it depends on the conditions we create around the instance for which we want to explain the black-box prediction.

9.2.5 A theoretical challenge

During the thesis, we identified an interesting theoretical question that is related to the usage of the product t-norm. Namely, in Chapter 6 we showed how to efficiently calculate granular approximations and how to prove their desirable properties when the product t-norm is used. However, due to the fact that the corresponding induced negator is not involutive, product t-norm was not used in the classification tasks. One can try to explore which types of problems are suitable for the product t-norm, i.e., which problems do not require an involutive negator. One possibility is that if the concept of disjoint granules is relaxed (i.e., the t-norm value is not exactly 0 but smaller than a positive threshold value), maybe the product t-norm and other similar t-norms can be used.

Bibliography

[1] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. *Network flows*. Prentice-Hall, Inc., 1993.

[2] Ravindra K Ahuja and James B Orlin. A capacity scaling algorithm for the constrained maximum flow problem. *Networks*, 25(2):89–98, 1995.

[3] Mark A Aizerman. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964.

[4] Mudabbir Ali, Asad Masood Khattak, Zain Ali, Bashir Hayat, Muhammad Idrees, Zeeshan Pervez, Kashif Rizwan, Tae-Eung Sung, and Ki-Il Kim. Estimation and interpretation of machine learning models with customized surrogate model. *Electronics*, 10(23):3045, 2021.

[5] Jose M Alonso, Ciro Castiello, and Corrado Mencar. Interpretability of fuzzy systems: Current research trends and prospects. *Springer handbook of computational intelligence*, pages 219–237, 2015.

[6] Claudi Alsina, Berthold Schweizer, and Maurice J Frank. *Associative functions: triangular norms and copulas*. World Scientific, 2006.

[7] Claudi Alsina and M Santos Tomás. Smooth convex t-norms do not exist. *Proceedings of the American Mathematical Society*, pages 317–320, 1988.

[8] Daniel W Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal*

of the Royal Statistical Society: Series B (Statistical Methodology), 82(4):1059–1086, 2020.

[9] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*, 2019.

[10] Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online, November 2020. Association for Computational Linguistics.

[11] Andrzej Bargiela and Witold Pedrycz. The roots of granular computing. In *2006 IEEE International Conference on Granular Computing*, pages 806–809. IEEE, 2006.

[12] Richard E Barlow and Hugh D Brunk. The isotonic regression problem and its dual. *Journal of the American Statistical Association*, 67(337):140–147, 1972.

[13] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Nozza Debora, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, Manuela Sanguinetti, et al. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *13th International Workshop on Semantic Evaluation*, pages 54–63. Association for Computational Linguistics, 2019.

[14] Bichitrananda Behera and G Kumaravelan. Text document classification using fuzzy rough set based on robust nearest neighbor (frs-rnn). *Soft Computing*, 25(15):9915–9923, 2021.

[15] Alessio Benavoli, Giorgio Corani, and Francesca Mangili. Should we really use post-hoc tests based on mean-ranks? *The Journal of Machine Learning Research*, 17(1):152–161, 2016.

[16] James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

[17] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Springer, 2006.

[18] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[19] José-Ramón Cano, Pedro Antonio Gutiérrez, Bartosz Krawczyk, Michał Woźniak, and Salvador García. Monotonic classification: An overview on algorithms, performance measures and data sets. *Neurocomputing*, 341:168–182, 2019.

[20] Kim Cao-Van and Bernard De Baets. An instance-based algorithm for learning rankings. In *Proceedings of Benelearn*, pages 15–21, 2004.

[21] Jorge Casillas, Oscar Cordón, Francisco Herrera, and Luis Magdalena. Interpretability improvements to find the balance interpretability-accuracy in fuzzy modeling: an overview. *Interpretability issues in fuzzy modeling*, pages 3–22, 2003.

[22] Ramaswamy Chandrasekaran, Young U Ryu, Varghese S Jacob, and Sungchul Hong. Isotonic separation. *INFORMS Journal on Computing*, 17(4):462–474, 2005.

[23] William W Cohen. Fast effective rule induction. In *Machine learning proceedings 1995*, pages 115–123. Elsevier, 1995.

[24] R Dennis Cook. Detection of influential observation in linear regression. *Technometrics*, 19(1):15–18, 1977.

[25] Chris Cornelis, Martine De Cock, and Anna Maria Radzikowska. Vaguely quantified rough sets. In *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, pages 87–94. Springer, 2007.

[26] Chris Cornelis, Richard Jensen, Germán Hurtado, Dominik Śle, et al. Attribute selection with fuzzy decision reducts. *Information Sciences*, 180(2):209–224, 2010.

[27] Chris Cornelis, Nele Verbiest, and Richard Jensen. Ordered weighted average based fuzzy rough sets. In *Proceedings of the 5th International Conference on Rough Sets and Knowledge Technology (RSKT 2010)*, pages 78–85, 2010.

[28] Chris Cornelis, Nele Verbiest, and Richard Jensen. Ordered weighted average based fuzzy rough sets. In *International Conference on Rough Sets and Knowledge Technology*, pages 78–85. Springer, 2010.

[29] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[30] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. Multi-objective counterfactual explanations. In *International Conference on Parallel Problem Solving from Nature*, pages 448–469. Springer, 2020.

[31] Trevor J Davis and C Peter Keller. Modelling uncertainty in natural resource analysis using fuzzy sets and monte carlo simulation: slope stability prediction. *International Journal of Geographical Information Science*, 11(5):409–434, 1997.

[32] Bernard De Baets and Radko Mesiar. Pseudo-metrics and t-equivalences. *Journal of Fuzzy Mathematics*, 5:471–481, 1997.

[33] Bernard De Baets and Radko Mesiar. Metrics and t-equalities. *Journal of mathematical analysis and applications*, 267(2):531–547, 2002.

[34] Lynn D'eer, Chris Cornelis, and Yiyu Yao. A semantical approach to rough sets and dominance-based rough sets. In *Proceedings of 16th International Conference on Information Processing and Management of Uncertainty (IPMU2016), Part II, CCIS 611*, pages 23–35, 2016.

[35] Lynn D'eer, Nele Verbiest, Chris Cornelis, and Lluís Godo. A comprehensive study of implicator–conjunctor-based and noise-tolerant fuzzy rough sets: definitions, properties and robustness analysis. *Fuzzy Sets and Systems*, 275:1–38, 2015.

[36] Chen Degang, Yang Yongping, and Wang Hui. Granular computing based on fuzzy similarity relations. *Soft Computing*, 15(6):1161–1172, 2011.

[37] Krzysztof Dembczyński, Wojciech Kotłowski, Salvatore Greco, and Roman Słowiński. Ensemble of decision rules for ordinal classification with monotonicity constraints. In *G.Wang et al. (eds.), Rough Sets and Knowledge Technology (RSKT 2008). LNAI 5009*, pages 260–267. Springer, Belin, 2008.

[38] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[39] Wen Sheng Du and Bao Qing Hu. Dominance-based rough fuzzy set approach and its application to rule induction. *European Journal of Operational Research*, 261(2):690–703, 2017.

[40] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[41] Didier Dubois and Henri Prade. Rough fuzzy sets and fuzzy rough sets. *International Journal of General System*, 17(2-3):191–209, 1990.

[42] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on machine learning models. *arXiv preprint arXiv:1707.08945*, 2(3):4, 2017.

[43] Tuan-Fang Fan, Churn-Jung Liau, and Duen-Ren Liu. Variable consistency and variable precision models for dominance-based fuzzy rough set analysis of possibilistic information systems. *International Journal of General Systems*, 42(6):659–686, 2013.

[44] Bo Wen Fang and Bao Qing Hu. Granular fuzzy rough sets based on fuzzy implicators and coimplicators. *Fuzzy Sets and Systems*, 359:112–139, 2019.

[45] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, 20(177):1–81, 2019.

[46] Evelyn Fix and Joseph Lawson Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247, 1989.

[47] David A Freedman. *Statistical models: theory and practice*. cambridge university press, 2009.

[48] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[49] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[50] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701, 1937.

[51] Susan Garavaglia and Asha Sharma. A smart guide to dummy variables: Four applications and a macro. In *Proceedings of the northeast SAS users group conference*, volume 43, 1998.

[52] Saul I Gass. *Linear programming: methods and applications*. Courier Corporation, 2003.

[53] Warren Gilchrist. *Statistical modelling with quantile functions*. CRC Press, 2000.

[54] Salvatore Greco, Masahiro Inuiguchi, and Roman Słowiński. Dominance-based rough set approach using possibility and necessity measures. In *International Conference on Rough Sets and Current Trends in Computing*, pages 85–92. Springer, 2002.

[55] Salvatore Greco, Benedetto Matarazzo, and Roman Slowinski. Fuzzy extension of the rough set approach to multicriteria and multiattribute sorting. In *Preferences and decisions under incomplete knowledge*, pages 131–151. Springer, 2000.

[56] Salvatore Greco, Benedetto Matarazzo, and Roman Slowinski. Rough sets theory for multicriteria decision analysis. *European journal of operational research*, 129(1):1–47, 2001.

[57] Salvatore Greco, Benedetto Matarazzo, and Roman Słowiński. Dominance-based rough set approach as a proper way of handling graduality in rough set theory. In *Transactions on rough sets VII*, pages 36–52. Springer, 2007.

[58] Salvatore Greco, Benedetto Matarazzo, and Roman Słowiński. Fuzzy set extensions of the dominance-based rough set approach. In *Fuzzy sets and their extensions: representation, aggregation and models*, pages 239–261. Springer, 2008.

[59] Salvatore Greco, Benedetto Matarazzo, and Roman Słowiński. The bipolar complemented de Morgan Brouwer-Zadeh distributive lattice as an algebraic structure for the dominance-based rough set approach. *Fundamenta Informaticae*, 115(1):25–56, 2012.

[60] Salvatore Greco, Benedetto Matarazzo, Roman Slowinski, and Jerzy Stefanowski. An algorithm for induction of decision rules consistent with the dominance principle. In *International Conference on Rough Sets and Current Trends in Computing*, pages 304–313. Springer, 2000.

[61] Salvatore Greco, Benedetto Matarazzo, Roman Słowiński, and Jerzy Stefanowski. Variable consistency model of dominance-based rough sets approach. In *International Conference on Rough Sets and Current Trends in Computing*, pages 170–181. Springer, 2000.

[62] Salvatore Greco, Benedetto Matarazzo, and Roman Słowiński. A new rough set approach to evaluation of bankruptcy risk. In *Operational tools in the management of financial risks*, pages 121–136. Springer, 1998.

[63] Salvatore Greco, Benedetto Matarazzo, and Roman Słowiński. Dominance-based rough set approach to granular computing. In *Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation*, pages 439–496. IGI Global, 2010.

[64] Brandon M Greenwell, Bradley C Boehmke, and Andrew J McCarthy. A simple and effective model-based variable importance measure. *arXiv preprint arXiv:1805.04755*, 2018.

[65] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[66] Jerzy W Grzymala-Busse. LERS-a system for learning from examples based on rough sets. In *Intelligent decision support*, pages 3–18. Springer, 1992.

[67] Jerzy W Grzymala-Busse and Jerzy Stefanowski. Three discretization methods for rule induction. *International Journal of Intelligent Systems*, 16(1):29–38, 2001.

[68] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022.

[69] Pedro Antonio Gutiérrez, Maria Perez-Ortiz, Javier Sanchez-Monedero, Francisco Fernandez-Navarro, and Cesar Hervas-Martinez. Ordinal regression methods: survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):127–146, 2015.

[70] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[71] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[72] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[73] Franz Huber. *A Logical Introduction to Probability and Induction*. Oxford University Press, 2018.

[74] Jens Hühn and Eyke Hüllermeier. Furia: an algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery*, 19(3):293–319, 2009.

[75] Masahiro Inuiguchi and Yukihiro Yoshioka. Variable-precision dominance-based rough set approach. In *International Conference on Rough Sets and Current Trends in Computing*, pages 203–212. Springer, 2006.

[76] Johan Ludwig William Valdemar Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193, 1906.

[77] Richard Jensen and Chris Cornelis. Fuzzy-rough instance selection. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–7. IEEE, 2010.

[78] Richard Jensen and Chris Cornelis. Fuzzy-rough nearest neighbour classification and prediction. *Theoretical Computer Science*, 412(42):5871–5884, 2011.

[79] Richard Jensen, Chris Cornelis, and Qiang Shen. Hybrid fuzzy-rough rule induction and feature selection. In *2009 IEEE International Conference on Fuzzy Systems*, pages 1151–1156. IEEE, 2009.

[80] Robert Ivor John and Peter R Innocent. Modeling uncertainty in clinical diagnosis using fuzzy logic. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(6):1340–1350, 2005.

[81] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th annual international symposium on computer architecture*, pages 1–12, 2017.

[82] Panagiota Karatza, Kalliopi Dalakleidi, Maria Athanasiou, and Konstantina S Nikita. Interpretability methods of machine learning algorithms with applications in breast cancer diagnosis. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 2310–2313. IEEE, 2021.

[83] Erich Peter Klement, Maddalena Manzi, Radko Mesiar, et al. Ultramodularity and copulas. *Rocky Mountain Journal of Mathematics*, 44(1):189–202, 2014.

[84] Erich Peter Klement, Radko Mesiar, and Endre Pap. *Triangular norms*, volume 8. Springer Science & Business Media, 2013.

[85] Roger Koenker and Kevin F Hallock. Quantile regression. *Journal of Economic Perspectives*, 15(4):143–156, 2001.

[86] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

[87] Teuvo Kohonen. Learning vector quantization. In *Self-organizing maps*, pages 175–189. Springer, 1995.

[88] Wojciech Kotłowski, Krzysztof Dembczyński, Salvatore Greco, and Roman Słowiński. Stochastic dominance-based rough set model for ordinal classification. *Information Sciences*, 178(21):4019–4037, 2008.

[89] Wojciech Kotłowski and Roman Słowiński. Statistical approach to ordinal classification with monotonicity constraints. In *Preference Learning ECML/PKDD 2008 Workshop*, 2008.

[90] John R Koza, Forrest H Bennett, David Andre, and Martin A Keane. Automated design of both the topology and sizing of analog electrical circuits using genetic programming. In *Artificial Intelligence in Design'96*, pages 151–170. Springer, 1996.

[91] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[92] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[93] ZhanFeng Liu and Su Pan. Fuzzy-rough instance selection combined with effective classifiers in credit scoring. *Neural Processing Letters*, 47(1):193–202, 2018.

[94] Prasanta Chandra Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, page 49––55, 1936.

[95] Jiří Matoušek. On directional convexity. *Discrete & Computational Geometry*, 25(3):389–403, 2001.

[96] Jiri Matousek and Bernd Gärtner. *Understanding and using linear programming*. Springer Science & Business Media, 2007.

[97] Alicja Mieszkowicz-Rolka and Leszek Rolka. Variable precision fuzzy rough sets. In *Transactions on Rough Sets I*, pages 144–160. Springer, 2004.

[98] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[99] Bernhard Moser. On the t-transitivity of kernels. *Fuzzy Sets and Systems*, 157(13):1787–1796, 2006.

[100] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[101] Kyoung-Su Oh and Keechul Jung. Gpu implementation of neural networks. *Pattern Recognition*, 37(6):1311–1314, 2004.

[102] Ibrahim Özkan and I Burhan Türkşen. Uncertainty and fuzzy decisions. In *Chaos Theory in Politics*, pages 17–27. Springer, 2014.

[103] Zdzisław Pawlak. Rough sets. *International journal of computer & information sciences*, 11(5):341–356, 1982.

[104] Witold Pedrycz. Allocation of information granularity in optimization and decision-making models: towards building the foundations of granular computing. *European Journal of Operational Research*, 232(1):137–145, 2014.

[105] Direnc Pekaslan, Chao Chen, Christian Wagner, and Jonathan M Garibaldi. Performance and interpretability in fuzzy logic systems–can we have both? In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 571–584. Springer, 2020.

[106] Rob Potharst and Adrianus Johannes Feelders. Classification trees for problems with monotonicity constraints. *ACM SIGKDD Explorations Newsletter*, 4(1):1–10, 2002.

[107] Madan L Puri and Dan A Ralescu. Fuzzy random variables. *Journal of Mathematical Analysis and Applications*, 114(2):409–422, 1986.

[108] Yuhua Qian, Qi Wang, Honghong Cheng, Jiye Liang, and Chuangyin Dang. Fuzzy-rough feature selection accelerator. *Fuzzy Sets and Systems*, 258:61–78, 2015.

[109] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[110] Enislay Ramentol, Sarah Vluymans, Nele Verbiest, Yailé Caballero, Rafael Bello, Chris Cornelis, and Francisco Herrera. Ifrowann: imbalanced fuzzy-rough ordered weighted average nearest neighbor classification. *IEEE Transactions on Fuzzy Systems*, 23(5):1622–1637, 2014.

[111] Enislay Ramentol, Sarah Vluymans, Nele Verbiest, Yailé Caballero, Rafael Bello, Chris Cornelis, and Francisco Herrera. Ifrowann: imbalanced fuzzy-rough ordered weighted average nearest neighbor classification. *IEEE Transactions on Fuzzy Systems*, 23(5):1622–1637, 2015.

[112] Alvin C Rencher and G Bruce Schaalje. *Linear models in statistics*. John Wiley & Sons, 2008.

[113] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[114] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016.

[115] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[116] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[117] Walter Rudin. *Real and Complex Analysis P. 2*. McGraw-Hill, 1970.

[118] Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1964.

[119] Stefan Schaal and Christopher G Atkeson. Robot juggling: implementation of memory-based learning. *IEEE Control Systems Magazine*, 14(1):57–71, 1994.

[120] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[121] Moshe Shaked and J George Shanthikumar. *Stochastic orders*. Springer Science & Business Media, 2007.

[122] Lloyd S Shapley. A value for n-person games. *Classics in game theory*, 69, 1997.

[123] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.

[124] Guido Van Rossum and Fred L Drake Jr. *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[125] Sarah Vluymans. *Dealing with imbalanced and weakly labelled data in machine learning using fuzzy and rough set methods*. PhD thesis, Springer, 2018.

[126] Sarah Vluymans, Chris Cornelis, Francisco Herrera, and Yvan Saeys. Multi-label classification using a fuzzy rough neighborhood consensus. *Information Sciences*, 433:96–114, 2018.

[127] Sarah Vluymans, Neil Mac Parthalain, Chris Cornelis, and Yvan Saeys. Weight selection strategies for ordered weighted average based fuzzy rough sets. *Information Sciences*, 501:155–171, 2019.

[128] Sarah Vluymans, Dánel Sánchez Tarragó, Yvan Saeys, Chris Cornelis, and Francisco Herrera. Fuzzy rough classifiers for class imbalanced multi-instance data. *Pattern Recognition*, 53:36–45, 2016.

[129] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[130] Jihong Wan, Hongmei Chen, Tianrui Li, Xiaoling Yang, and Binbin Sang. Dynamic interaction feature selection based on fuzzy rough set. *Information Sciences*, pages 891–911, 2021.

[131] Changzhong Wang, Yang Huang, Weiping Ding, and Zehong Cao. Attribute reduction with fuzzy rough self-information measures. *Information Sciences*, 549:68–86, 2021.

[132] Chun Yong Wang and Bao Qing Hu. Granular variable precision fuzzy rough sets with general fuzzy relations. *Fuzzy Sets and Systems*, 275:39–57, 2015.

[133] Philip Wolfe. The simplex method for quadratic programming. *Econometrica: Journal of the Econometric Society*, pages 382–398, 1959.

[134] Abraham J Wyner, Matthew Olson, Justin Bleich, and David Mease. Explaining the success of adaboost and random forests as interpolating classifiers. *The Journal of Machine Learning Research*, 18(1):1558–1590, 2017.

[135] Ronald R Yager. Set-based representations of conjunctive and disjunctive knowledge. *Information Sciences*, 41(1):1–22, 1987.

[136] Ronald R Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on systems, Man, and Cybernetics*, 18(1):183–190, 1988.

[137] Jing Tao Yao, Athanasios V Vasilakos, and Witold Pedrycz. Granular computing: perspectives and challenges. *IEEE Transactions on Cybernetics*, 43(6):1977–1989, 2013.

[138] Y Yao. Rough sets, neighborhood systems and granular computing. In *Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No. 99TH8411)*, volume 3, pages 1553–1558. IEEE, 1999.

[139] Yanqing Yao, Jusheng Mi, and Zhoujun Li. A novel variable precision (θ, σ)-fuzzy rough set model based on fuzzy granules. *Fuzzy Sets and Systems*, 236:58–72, 2014.

[140] YY Yao. Granular computing using neighborhood systems. In *Advances in soft computing*, pages 539–553. Springer, 1999.

[141] YY Yao. Information granulation and rough set approximation. *International Journal of Intelligent Systems*, 16(1):87–104, 2001.

[142] Zhong Yuan, Hongmei Chen, Tianrui Li, Zeng Yu, Binbin Sang, and Chuan Luo. Unsupervised attribute reduction for mixed data based on fuzzy rough sets. *Information Sciences*, 572:67–87, 2021.

[143] Zhong Yuan, Hongmei Chen, Peng Xie, Pengfei Zhang, Jia Liu, and Tianrui Li. Attribute reduction methods in fuzzy rough set theory: An overview, comparative experiments, and new directions. *Applied Soft Computing*, 107:107353, 2021.

[144] Lotfi Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[145] Lotfi A Zadeh. Fuzzy sets and information granularity. *Advances in fuzzy set theory and applications*, 11:3–18, 1979.

[146] Lotfi A Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy sets and systems*, 90(2):111–127, 1997.

[147] Lotfi Asker Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1):3–28, 1978.

[148] Suyun Zhao, Zhigang Dai, Xizhao Wang, Peng Ni, Hengheng Luo, Hong Chen, and Cuiping Li. An accelerator for rule induction in fuzzy rough theory. *IEEE Transactions on Fuzzy Systems*, 29(12):3635–3649, 2021.

[149] Suyun Zhao, Eric CC Tsang, Degang Chen, and Xizhao Wang. Building a rule-based classifier—a fuzzy-rough set approach. *IEEE Transactions on Knowledge and Data Engineering*, 22(5):624–638, 2009.

[150] Shang-Ming Zhou and John Q Gan. Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling. *Fuzzy sets and systems*, 159(23):3091–3131, 2008.

List of publications

Papers in international journals listed in the Science Citation Index

- Marko Palangetić, Chris Cornelis, Salvatore Greco, and Roman Słowiński. Fuzzy granular approximation classifier. *arXiv preprint arXiv:2206.01240*, 2022
- Marko Palangetić, Chris Cornelis, Salvatore Greco, and Roman Słowiński. Multi-class granular approximation by means of disjoint and adjacent fuzzy granules. *arXiv preprint arXiv:2202.07584*, 2022
- Marko Palangetić, Chris Cornelis, Salvatore Greco, and Roman Słowiński. A novel machine learning approach to data inconsistency with respect to a fuzzy relation. *arXiv preprint arXiv:2111.13447*, 2021
- Marko Palangetić, Chris Cornelis, Salvatore Greco, and Roman Słowiński. Granular representation of OWA-based fuzzy rough sets. *Fuzzy Sets and Systems*, 440:112–130, 2022
- Marko Palangetić, Chris Cornelis, Salvatore Greco, and Roman Słowiński. Fuzzy extensions of the dominance-based rough set approach. *International Journal of Approximate Reasoning*, 129:1–19, 2021

Conference proceedings

- Marko Palangetić, Chris Cornelis, Salvatore Greco, and Roman Słowiński. Rough sets meet statistics-a new view on rough set rea-

soning about numerical data. In *International Joint Conference on Rough Sets*, pages 78–92. Springer, 2020

- Marko Palangetić, Chris Cornelis, Salvatore Greco, and Roman Słowiński. Extension of the fuzzy dominance-based rough set approach using ordered weighted average operators. In *11th Conference of the European-Society-for-Fuzzy-Logic-and-Technology (EUSFLAT)*, volume 1, pages 528–535. Atlantis Press, 2019