Information Sciences 629 (2023) 249–275

Contents lists available at ScienceDirect

Information Sciences

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

Granular approximations: A novel statistical learning approach for handling data inconsistency with respect to a fuzzy relation

Marko Palangetić [a,*], Chris Cornelis [a], Salvatore Greco [b,c], Roman Słowiński [d,e]

[a] Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Ghent, Belgium
[b] Department of Economics and Business, University of Catania, Catania, Italy
[c] Portsmouth Business School, Centre of Operations Research and Logistics (CORL), University of Portsmouth, Portsmouth, United Kingdom
[d] Institute of Computing Science, Poznań University of Technology, Poznań, Poland,
[e] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

ARTICLE INFO

Keywords:
Inconsistencies in data
Fuzzy logic
Statistical learning
Rough sets

ABSTRACT

Inconsistency in classification and regression problems occurs when instances that relate in a certain way on the condition attributes, do not follow the same relation on the decision attribute. It typically appears as a result of perturbation in data caused by incomplete knowledge (missing attributes) or by random effects that occur during data generation (instability in the assessment of decision attribute values). Inconsistencies with respect to a crisp preorder relation (expressing either dominance or indiscernibility between instances) can be handled with set-theoretic approaches like rough sets and by using statistical/machine learning approaches that involve optimization methods. In particular, the Kotłowski-Słowiński (KS) approach relabels the objects from a dataset such that inconsistencies are removed, and such that the new class labels are as close as possible to the original ones in terms of a given loss function. In this paper, we generalize the KS approach to handle inconsistency determined by a fuzzy preorder relation rather than a crisp one. The method produces a consistent fuzzy relabeling of the instances and may be used as a preprocessing tool with algorithms for binary classification and regression. As the obtained fuzzy sets can be represented as unions of meaningful simple fuzzy sets or granules, we call them granular approximations. We provide statistical foundations for our method, develop appropriate optimization procedures, provide didactic examples, and prove several important properties.

1. Introduction

Ordinal classification (also called ordinal regression) problems constitute a very important part of machine learning and statistical analysis [1,2]. In ordinal classification, the goal is to predict for a certain instance u from set U, one of K different ordinal class labels $y \in \{1, \ldots, K\}$. Usually, u is characterized by its values for a given set of condition attributes, while y is called a decision attribute. Ordinal classification problems take into account the existing ordering on the decision attribute. In some cases, an ordering also exists on the condition attributes. One way to incorporate such knowledge in the analysis is through so-called monotonicity constraints. For

* Corresponding author.
E-mail addresses: marko.palangetic@ugent.be (M. Palangetić), chris.cornelis@ugent.be (C. Cornelis), salgreco@unict.it (S. Greco), roman.slowinski@cs.put.poznan.pl (R. Słowiński).

https://doi.org/10.1016/j.ins.2023.01.119
Received 25 November 2021; Received in revised form 27 January 2023; Accepted 28 January 2023

Available online 1 February 2023

a given preorder (dominance) relation on the set of instances U based on the condition attributes, the monotonicity constraints can be formulated as follows: if instance u_1 dominates (is dominated by) u_2 on the condition attributes, then u_1 should be assigned to the same or to a better (to a worse) decision class than u_2. In this case, we say that u_1 is consistent with u_2 w.r.t. the dominance relation. Obviously, consistency is a symmetric relation, and instances that are incomparable on condition attributes are consistent by default.

Ordinal classification problems that include monotonicity constraints are called monotone classification problems. They emerge in many areas, such as medical diagnosis [3], bankruptcy risk estimation [4], house pricing [5], and others. A comprehensive survey of monotone classification methods is given in [6].

In practice, not all pairs of instances satisfy the monotonicity constraints (are not consistent) due to some imperfections of ordinal classification data, like missing attributes or instability of the assessment of decision attribute values at the data generation stage. One way of dealing with this problem is by adapting machine learning algorithms to mitigate the effect of monotonicity constraint violations, as was done in [7] for monotonic fuzzy nearest neighbor classification. One may also opt to generalize the label space, such that each instance is assigned a superset of possible labels, as was considered e.g. in [8]. An application-independent approach that operates in the original label space is to perform some initial preprocessing of the data to explicitly enforce the monotonicity constraints. The most basic and intuitive method to handle inconsistencies in data is the rough set approach [9]. For a given decision class in a classification problem, the approach outputs lower and upper approximations of that class. The lower approximation contains instances from the decision class that are consistent with all other instances, while the upper approximation contains instances that relate to instances from the decision class. The original rough set approach handles inconsistencies w.r.t. an equivalence (indiscernibility) relation. To make it applicable to monotone classification problems, Greco et al. [10] extended the original rough set theory with their Dominance-based Rough Set Approach (DRSA) which replaces the indiscernibility relation with a dominance relation. Since the introduction of DRSA, the original rough set theory is usually referred to as Indiscernibility-based Rough Set Approach (IRSA). Recently, the two were integrated into the Preorder-based Rough Set Approach (PRSA) [11].

The main limitation of the existing methods, which are designed in the scope of rough set theory, is that they eliminate inconsistencies in an extreme way through the lower and upper approximations. All inconsistent pairs are either removed from the decision class (lower approximation) or kept in the decision class (upper approximation).

A more comprehensive analysis of monotone classification from the statistical learning point of view was given by Kotłowski and Słowiński [12]. They provided statistical foundations of the monotonicity constraints and developed a machine learning method to incorporate them into data analysis. This method removes inconsistencies in data ("monotonizes" them) as a result of an optimization procedure that minimizes the cost of label changes in the decision attribute. It produces a new set of labels called a *monotone approximation*. This approach generalizes standard rough sets, provides a probabilistic view of them, and corrects inconsistencies in a non-extreme and theoretically optimal way. The approach found its application in the same areas as DRSA [13], as well as in the development of rule induction and ensemble rules methods [14]. A well-known case of the approach is the isotonic regression model [15] which uses weighted mean squared error as its empirical risk. In the remainder of the paper, we refer to this method as *KS approach*.

On the other hand, fuzzy logic studies the gradual truth of logical statements, and is used extensively in modeling imprecise and vague information. Two ways to utilize fuzzy logic in data analysis are through fuzzy relations and fuzzy membership degrees in decision classes. Fuzzy relations are able to model relationships between numerical vectors. This is suitable to model similarity between numerical vectors or other structures (graphs, strings, DNA chains ...).

Fuzzy membership degrees allow that an instance can belong to multiple decision classes to different degrees. This is important in applications where the observed decisions are crisp, while the actual decisions exhibit fuzzy graduality. An example is a movie streaming service, where users grade movies in a binary way ("like" or "dislike"), while the actual information is gradual since a user may prefer one movie over another despite giving them both a "like" grade.

The integration of fuzzy logic and IRSA was first proposed by Dubois and Prade [16], allowing to approximate fuzzy sets using a fuzzy similarity relation. A similar extension of DRSA to fuzzy set theory was proposed by Greco et al. [17].

This article is motivated by the KS approach. Its main innovation is 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 now consider a fuzzy T-preorder relation to model the relationship between different instances on the condition attributes, where T refers to a given t-norm that models conjunction in fuzzy logic. On the other hand, in our approach, the decision attribute is represented as a fuzzy set, i.e., it takes values from the interval $[0, 1]$. Hence, it is appropriate for problems where the decision attribute can be modeled using values from this interval; concretely, for binary classification and regression problems. Our proposal generalizes the KS approach for the case of binary classification, and for specific loss functions (in particular, squared error loss and quantile loss). It also generalizes the lower and upper approximation from fuzzy rough set theory, as they are obtained as special cases when the quantile loss function is used.

Just like the KS approach [12], our proposal is also interesting from the granular computing point of view. Granular computing is a paradigm that involves a partition of information into meaningful groups, classes, or clusters called granules [18,19], and which has been applied to diverse models in data analysis. For example, in [20] and [21], granular computing using neighborhood systems for the interpretation of granules was studied, while in [22,23], granular aspects of rough set theory were examined.

In particular, the sets obtained with the KS approach, as well as with the novel approach, possess the property of granular representation: they can be represented as unions of meaningful granules [11,24]. Such sets are called granularly representable [11]. Due to the granular properties of our new approach, we call its result a *granular approximation*.

What distinguishes our method from those based on fuzzy rough sets is that it corrects inconsistency in an optimal way, producing granular approximations that are minimally different from the approximated fuzzy set (w.r.t. the adopted measure of difference,

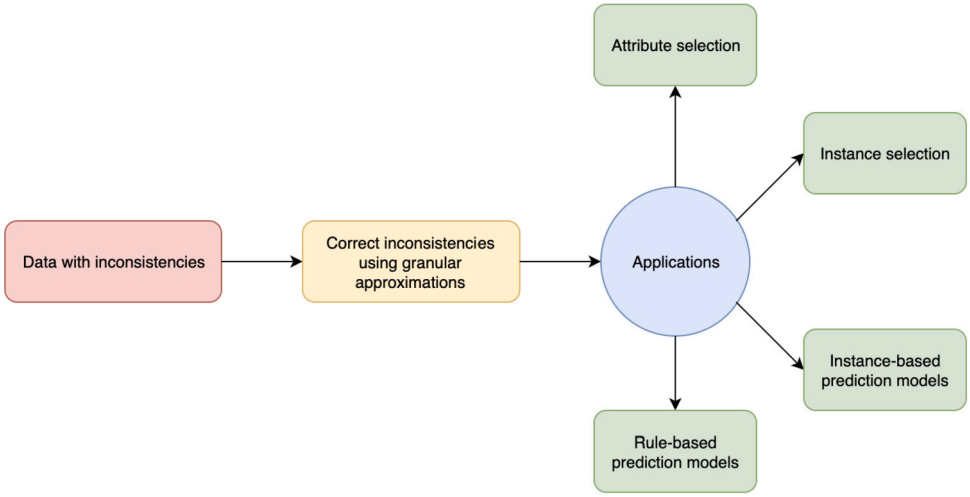

Fig. 1. Possible applications of the granular approximations.

which, in our case, is a loss function), whereas with fuzzy rough sets, the approximations are defined explicitly without minimizing any measure of difference between the approximations and the original fuzzy set.

In Fig. 1, we position our contribution w.r.t. potential applications. Correcting inconsistency by means of our method can be seen as a data preprocessing step, and the granular approximation may be used in the same applications as those where fuzzy rough sets are used. For example, fuzzy rough sets are applied in fuzzy rule induction methods [25,26], instance-based methods [27,28], instance selection methods [29] and attribute selection methods [30].

This paper is also an example of the successful integration of ideas and contributions of rough sets, fuzzy sets, and machine learning. Handling inconsistency and granulation are the main contributions of rough sets. The theory of fuzzy sets allows us to use fuzzy relations to model a non-binary interaction among instances. 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 remainder of the paper is organized as follows. In Section 2, we recall the required preliminaries about statistical learning theory, monotone approximations, fuzzy logic, and fuzzy rough sets. In Section 3, we define inconsistencies and illustrate how they occur in real data by didactic example. In Section 4, we develop the statistical foundations of granular approximations. Section 5 deals with the optimization problems that output granular approximations, while some important properties are proven in Section 6. Section 7 contains our conclusion and outlines future work.

In Appendix A–B, we deal with the dual formulations of the optimization problems introduced in Section 5. Using the duality theory, we obtain greedy algorithms for the optimization problems from Section 5 that allow us to prove Proposition 6.3.

2. Preliminaries

2.1. 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 $[0,1]$. 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 [31]:

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

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 2.1. *[32] 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 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 wish 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 by h^*, is called the Bayes predictor. The relationship between $\mathcal{X}$ and $\mathcal{Y}$ may be represented by a family of random variables $\mathcal{Y}_{\mathcal{X}=x}$, which stands for variable $\mathcal{Y}$ conditioned on $\mathcal{X} = x$. 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 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)

$$L(y, \hat{y}) = (y - \hat{y})^2, \tag{2}$$

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 absolute error loss

$$L(y, \hat{y}) = |y - \hat{y}|,$$

then the Bayes predictor is $h^*(x) = Q_{\mathcal{Y}|\mathcal{X}=x}(\frac{1}{2})$, i.e., the conditional median [33]. The previous examples show that a Bayes predictor is a characteristic of family $\mathcal{Y}_{\mathcal{X}=x}$ (conditional mean and median in the examples).

In practice, the random variables $\mathcal{X}$ and $\mathcal{Y}$ are unknown and we only have their realizations $x_1, \dots, x_n$ and $y_1, \dots, y_n$. Our goal is then to minimize the empirical risk:

$$\hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, h(x_i)). \tag{3}$$

Minimization of the empirical risk is one form of *learning* and it basically amounts to an estimation of the unknown Bayes predictor. The empirical risk for the squared error loss is called the mean squared error, while for the absolute error loss is called the mean absolute error. Also, since multiplying an objective function with a positive constant does not change the solution, factor $\frac{1}{n}$ is often omitted in (3). 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 [12] introduced a statistical framework for monotone classification. In this case, it is assumed that there is a preorder 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 $1, \dots, K$. 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 $K_{-1} = \{1, \dots, K-1\}$. In probabilistic terms, the monotonicity constraint says that $x \succeq x'$ implies

$$\begin{aligned} & \forall k \in K_{-1},\ P(\mathcal{Y} \leq k | \mathcal{X} = x) \leq P(\mathcal{Y} \leq k | \mathcal{X} = x') \\ \Leftrightarrow\ & \forall k \in K_{-1},\ F_{\mathcal{Y}|\mathcal{X}=x}(k) \leq F_{\mathcal{Y}|\mathcal{X}=x'}(k) \\ \Leftrightarrow\ & \forall p \in (0,1),\ Q_{\mathcal{Y}|\mathcal{X}=x}(p) \geq Q_{\mathcal{Y}|\mathcal{X}=x'}(p). \end{aligned} \tag{4}$$

The previous expression means that the probability that x will be assigned to at most class k is smaller or equal to that x' will be assigned to the same class. A family $\mathcal{Y}_{\mathcal{X}=x}$ is *monotonically constrained* if (4) 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 [12] showed that both squared error loss and absolute error loss are monotone loss functions. They also examined a parametrized family of monotone loss functions called p-quantile loss defined as:

$$L_p(y, \hat{y}) = (y - \hat{y})(p - \mathbf{1}_{y-\hat{y}<0}) = \begin{cases} p|y - \hat{y}| & \text{if } y - \hat{y} > 0, \\ (1-p)|y - \hat{y}| & \text{otherwise,} \end{cases} \tag{5}$$

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. For the p-quantile loss, we have the following important result proved in [12].

Proposition 2.2. *Let $s : \mathcal{Y} \to \mathbb{R}$ be an increasing function. Then the loss functions $L_p(y, \hat{y})$ and $L_p(s(y), s(\hat{y}))$ have the same Bayes predictor.*

Proposition 2.2 states that a different scaling of ordinal classes does not change the Bayes predictor, only the order matters.

Table 1
Some common t-norms and their R-implicators.

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

2.2. 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 that 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

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{n} L(\bar{y}_i, \hat{y}_i) \\ &\text{subject to} && x_i \succeq x_j \implies \hat{y}_i \ge \hat{y}_j, \quad i,j = 1,\dots,n \\ & && \hat{y}_i \in \{1,\dots,K\}, \quad i = 1,\dots,n \end{aligned} \tag{6}$$

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 (6) can be solved using linear programming. Moreover, the solutions of the linear optimization problem will always be integers due to the unimodularity of the constraint matrix [34].

2.3. Fuzzy logic connectives

The definitions and terminology in this subsection are based on [35]. Recall that a *t-norm* $T : [0,1]^2 \to [0,1]$ 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 1.

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

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

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 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 1 is strict while the others are not.

We call a t-norm Archimedean if

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

T_P, T_L and T_D from Table 1 are Archimedean, while T_M and T_{nM} are not.

A t-norm is a continuous Archimedean t-norm if and only if it is either strict or nilpotent.

We say that two t-norms T_1 and T_2 are isomorphic if there exists a bijection $\varphi : [0,1] \to [0,1]$ such that $T_1 = \varphi^{-1}(T_2(\varphi(x), \varphi(y)))$.

Proposition 2.3. *A strict t-norm is isomorphic to T_P while a nilpotent t-norm is isomorphic to T_L.*

We denote with

$$T_{L,\varphi} = \varphi^{-1}(\max(\varphi(x) + \varphi(y) - 1, 0)) \tag{7}$$

a family of nilpotent t-norms, i.e., t-norms that are isomorphic to T_L with bijection φ and we denote with

$$T_{P,\varphi} = \varphi^{-1}(\varphi(x)\varphi(y)) \tag{8}$$

a family of strict t-norms, i.e., t-norms that are isomorphic to T_L with bijection φ.

We say that a t-norm is D-convex if its partial mappings are convex functions. T_P and T_L from Table 1 are D-convex, while the others are not. More details on the characterization of D-convex t-norms can be found in [11].

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 $I(0,0) = I(0,1) = I(1,1) = 1$.

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

$$T(x,y) \leq z \Leftrightarrow x \leq I(y,z). \tag{9}$$

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 1 shows the residual implicators of the corresponding t-norms. Note that all of them, except I_{T_D}, satisfy the residuation property. Implicators that satisfy the residuation principle have the ordering property

$$x \leq y \Leftrightarrow I(x,y) = 1. \tag{10}$$

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 $I_{P,\varphi}$.

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 left continuous t-norm T and its R-implicator I, we define the negator induced by I as $N(x) = I(x,0)$. We will call a triplet (T,I,N) obtained as previously explained a residual triplet. If a 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

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

Proposition 2.4. *For a residual triplet, the following holds:*

$$I(T(x,y),z) = I(x,I(y,z)).$$

As a consequence, when $z = 0$,

$$N(T(x,y)) = I(x,N(y)).$$

The standard negator is obtained when, for example, the t-norm is the Łukasiewicz one. In general, a t-norm for which the induced negator of its R-implicator is involutive is called an IMTL t-norm. We will call a residual triplet (T,I,N) that is generated with an IMTL t-norm, an IMTL triplet.

2.4. 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 a membership function that 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. 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 relation while a symmetric T-preorder relation is called a T-equivalence.

To recall some of these fuzzy relations, we assume that instances from U are described with a finite set of numerical attributes Q. Let $u^{(q)}$ and $v^{(q)}$ be the evaluations of instances u and v on attribute q. For a gain-type[1] attribute $q \in Q$, an example, given in [36], of a T_L-preorder relation (expressing dominance) on attribute q, is

[1] An attribute is considered to be gain-type when the higher its evaluation, the better it is considered to be. In the opposite case, we call it a cost-type attribute.

$$\widetilde{R}_q^{\gamma}(u,v) = \max\left(\min\left(1-\gamma\frac{v^{(q)}-u^{(q)}}{range(q)},1\right),0\right), \tag{11}$$

where γ is a positive parameter and $range(q)$ is the difference between the maximal and minimal value of q. For a cost-type attribute $q \in Q$, the analogous T_L-preorder relation is

$$\widetilde{R}_q^{\gamma}(u,v) = \max\left(\min\left(1-\gamma\frac{u^{(q)}-v^{(q)}}{range(q)},1\right),0\right). \tag{12}$$

The overall relation considering all attributes jointly may be defined as

$$\widetilde{R}^{\gamma}(u,v) = \min_{q\in Q}\widetilde{R}_q^{\gamma}(u,v). \tag{13}$$

An example of a T_L-equivalence relation is called triangular similarity, defined as

$$\widetilde{R}^{\gamma}(u,v) = \min_{q\in Q}\max\left(1-\gamma\frac{|u^{(q)}-v^{(q)}|}{range(q)},0\right). \tag{14}$$

More details on such similarity relations are provided in [36].

Finally, note that the introduced fuzzy relations depend on the parameter γ. If γ tends to infinity, these fuzzy relations are reduced to the usual crisp relations (indiscernibility and dominance relations, as discussed before). The value of parameter γ may depend on the use of inconsistency correction techniques. For example, if they are used in predictive machine learning models, γ can be considered as a model's hyperparameter and can be tuned using various techniques (for a state of the art survey on the subject see, e.g., [37]).

2.5. Fuzzy rough approximations and granular representability

Let U be a set of instances, A a fuzzy set on U, and let $\widetilde{R}$ be a T-preorder relation on U. The fuzzy PRSA lower and upper approximations of A are fuzzy sets for which the membership function is defined as:

$$\begin{aligned}\underline{\mathrm{apr}}_{\widetilde{R}}^{\min,I}(A)(u) &= \min\{I(\widetilde{R}(v,u),A(v)); v\in U\}\\ \overline{\mathrm{apr}}_{\widetilde{R}}^{\max,T}(A)(u) &= \max\{T(\widetilde{R}(u,v),A(v)); v\in U\},\end{aligned} \tag{15}$$

for $u \in U$. The approximations have some important properties [36]:

- (**inclusion**) $\underline{\mathrm{apr}}_{\widetilde{R}}^{\min,I}(A) \subseteq A \subseteq \overline{\mathrm{apr}}_{\widetilde{R}}^{\max,T}(A)$.
- (**duality**) $co\underline{\mathrm{apr}}_{\widetilde{R}}^{\min,I}(A) = \overline{\mathrm{apr}}_{\widetilde{R}}^{\max,T}(coA)$, $co\overline{\mathrm{apr}}_{\widetilde{R}}^{\max,T}(A) = \underline{\mathrm{apr}}_{\widetilde{R}}^{\min,I}(coA)$ when an IMTL triplet is used.
- (**consistency**) $\widetilde{R}(u,v) \le I(\underline{\mathrm{apr}}_{\widetilde{R}}^{\min,I}(A)(v), \underline{\mathrm{apr}}_{\widetilde{R}}^{\min,I}(A)(u))$,
  $\widetilde{R}(u,v) \le I(\overline{\mathrm{apr}}_{\widetilde{R}}^{\max,T}(A)(v), \overline{\mathrm{apr}}_{\widetilde{R}}^{\max,T}(A)(u))$.

Given $\lambda \in [0,1]$, T-preorder $\widetilde{R}$, t-norm T and $u \in U$, a fuzzy granule is defined as a parametric fuzzy set

$$\widetilde{R}_{\lambda}^{+}(u) = \{(v, T(\widetilde{R}(u,v),\lambda); v\in U\}. \tag{16}$$

A fuzzy set A in U is granularly representable w.r.t. $\widetilde{R}$ and T if

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

where the union of fuzzy sets is defined with the max operator.

Proposition 2.5. *[11] It holds that $\underline{apr}_{\widetilde{R}}^{\min,I}(A)$ is the largest granularly representable set contained in A, while $\overline{apr}_{\widetilde{R}}^{\max,T}(A)$ is the smallest granularly representable set containing A.*

We can observe the connection between consistency and granular representability by the fact that the lower and upper approximations possess both properties.

3. Inconsistencies in data - definition and didactic examples

In this section, we provide the formal definition of data inconsistency and illustrate it using two small examples. Let U be a set of instances, $\widetilde{R}$ 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

$$T(\widetilde{R}(u,v), A(v)) \le A(u), \tag{17}$$

or equivalently,

Table 2
Classification data.

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
T_L-equivalence matrix on classification data.

	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

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

Both forms are valid due to the residuation property (9). In order to better understand Eq. (17), 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. (17) 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. (17) 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 a complete absence of similarity. The interpretation of Eq. (17) 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. (17) 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.

3.1. Binary classification

Consider the binary classification problem in Table 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.

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 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 while the others are gain-type attributes). 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, we calculate pairwise similarities among instances from Table 2 using T_L-equivalence (14) for $\gamma = 1$. If we are dealing with a classification problem, 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.

In Table 4, we calculate the pairwise fuzzy dominance values among instances from Table 2 using T_L-preorder (13) for $\gamma = 1$. Using the same pair of instances, we can identify the inconsistency w.r.t. the fuzzy dominance relation.

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.

Table 4
T_L-preorder matrix on classification data.

	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 5
Regression data.

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 6
T_L-equivalence matrix on regression data.

	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

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 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. Regression

In the examples derived from the data from Table 2, inconsistencies w.r.t. a fuzzy relation were observed when we deal with a crisp decision. In Table 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.

Since we are dealing with fuzzy decision labels, it is not possible to consider inconsistencies w.r.t. a crisp relation. Therefore, we will identify inconsistencies w.r.t. fuzzy relations. Pairwise evaluations of the T_L-equivalence relation (14) on instances from Table 5 are given in Table 6.

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

Pairwise evaluations of the T-preorder relation (14) on instances from Table 5 are given in Table 7. Using the same pair of instances, we can identify the inconsistency w.r.t. the fuzzy dominance relation.

3.3. Relationship between consistency and granular representability

As already stated before, the methods that are used for inconsistency corrections (rough sets and KS approach) in the crisp case, as well as the fuzzy rough sets exhibit granular properties i.e., the resulting set without inconsistencies can be represented as a union of smaller meaningful sets called granules. Such properties can be called granular properties.

The following proposition reveals the equivalence between granular representability and consistency for a general T-preorder relation.

Table 7
T-preorder matrix on regression data.

	1	2	3	4	5	6
1	1.000	0.923	0.769	1.000	0.792	1.000
2	0.333	1.000	0.625	0.917	0.125	1.000
3	0.429	0.857	1.000	1.000	0.500	1.000
4	0.143	0.500	0.346	1.000	0.208	0.692
5	0.231	0.154	0.000	0.654	1.000	0.346
6	0.000	0.429	0.500	0.792	0.000	1.000

Proposition 3.1. *A fuzzy set A in U is granularly representable if and only if it satisfies the object monotonicity property, i.e.,*

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

Proof.

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

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. □

Due to the granular properties of data that are consistent w.r.t. a T-preorder, the term "monotonically constrained" can be translated into "consistent" or "granularly representable" introduced in [11] when dealing with T-preorder relations. We may use both terms interchangeably in our work. The following section reflects the consistency condition from a statistical point of view, i.e., from the assumption that data, including fuzzy decision attributes, are realizations of a random variable.

4. Statistical approach to inconsistency in data

4.1. Ontic fuzzy sets and probabilistic uncertainty

Before we proceed with the statistical view of 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 [38–40]. 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 [41]. 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 [42]. In this article, 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 is solely related to the unknown 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.

4.2. 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 the 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 Section 2.1, 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 [43] (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 Section 2. 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 (4) for a T-preorder relation. In order to relate granular representability and the family of random variables $\{\mathcal{A}(u), u \in U\}$, we introduce the following definition.

Definition 4.1. Random fuzzy set $\mathcal{A}$ is granularly representable (does not possess inconsistencies) if

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

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

Definition 4.1 is an extension of the third equivalence in (4). 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 [44].

Proposition 4.1. *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*

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

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

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

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

Proposition 4.2. *Let T be a D-convex t-norm and I its R-implicator. Then $E\mathcal{A}$ is granularly representable (does not possess inconsistencies) as soon as $\mathcal{A}$ is.*

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

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

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

$$\begin{aligned} T(\widetilde{R}(u,v), E\mathcal{A}(v)) &= T\left(\widetilde{R}(u,v), \int_0^1 A_p(v)\,dp\right) \\ &\le \int_0^1 T(\widetilde{R}(u,v), A_p(v))\,dp \\ &\le \int_0^1 A_p(u)\,dp = E\mathcal{A}(u). \end{aligned}$$

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. □

5. 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 loss function. For a given loss function L, the general form of the optimization problem expressing our goal is

$$\begin{aligned} &\text{minimize} && \sum_{u\in U} L(\bar{A}(u),\hat{A}(u)) \\ &\text{subject to} && T(\widetilde{R}(u,v),\hat{A}(v)) \le \hat{A}(u), \quad u,v\in U \\ & && 0\le \hat{A}(u)\le 1, \quad u\in U, \end{aligned} \tag{18}$$

where $\{\hat{A}(u), u\in U\}$ is the unknown granularly representable set. We will call the result of optimization problem (18) the *granular approximation* of fuzzy set $\{\bar{A}(u), u\in U\}$.

Optimization problem (18) is the main contribution of the article. 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 (18) 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 5.1. 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 (5) is granular, since its Bayes predictor is the quantile fuzzy set A_p, which is granularly representable by the definition. The squared error loss (2) is granular for D-convex t-norms since the Bayes predictor $E\mathcal{A}$ is granularly representable in this case by Proposition 4.2. Hence, both loss functions introduced in Subsection 2.1 are suitable for the calculation of granular approximations.

In problem (18), 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 the case of the loss functions (5) and (2), 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. (7) and (8). If $T_{L,\varphi}$ is used in (18), then the set of constraints that express granular representability can be simplified in the following way: for all $u,v\in U$,

$$\begin{aligned} &\widetilde{R}(v,u)\le I_{L,\varphi}(A(u),A(v)) \\ \Leftrightarrow\ & T_{L,\varphi}\left(\widetilde{R}(v,u),A(u)\right)\le A(v) \\ \Leftrightarrow\ & \varphi^{-1}(\max(\varphi(\widetilde{R}(u,v))+\varphi(\hat{A}(v))-1,0)\le \hat{A}(u) \\ \Leftrightarrow\ & \max(\varphi(\widetilde{R}(u,v))+\varphi(\hat{A}(v))-1,0)\le \varphi(\hat{A}(u)) \\ \Leftrightarrow\ & \max(\widetilde{R}_\varphi(u,v)+\alpha_v-1,0)\le \alpha_u \\ \Leftrightarrow\ & \widetilde{R}_\varphi(u,v)\le \alpha_u-\alpha_v+1 \end{aligned}$$

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

$$\varphi^{-1}(\varphi(\widetilde{R}(u,v))\varphi(\hat{A}(v)))\le\hat{A}(u)\Leftrightarrow\alpha_v\widetilde{R}_\varphi(u,v)\le\alpha_u$$

for all $u,v\in U$.

The border constraints now become $0\le\alpha_u\le 1$ 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 that can lead to a non-convex optimization problem. However, Proposition 2.2 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 apply φ to both parameters of the loss function and replace $L_p(\bar{A}(u),\varphi^{-1}(\alpha_u))$ by $L_p(\varphi(\bar{A}(u)),\alpha_u)$. Although the value of the estimand (the quantity that is estimated, i.e., the Bayes predictor A_p) remains unchanged with the new 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 quantile risk as a linear program [45]. 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 (18) as

$$\begin{aligned} \text{minimize} \quad & p \sum_{u \in U} x_u + (1-p) \sum_{u \in U} y_u, \\ \text{subject to} \quad & \alpha_u - \alpha_v + 1 \geq \widetilde{R}_\varphi(u, v), && u, v \in U \\ & x_u - y_u = \bar{A}_\varphi(u) - \alpha_u, && u \in U \\ & 0 \leq \alpha_u \leq 1,\ x_u \geq 0,\ y_u \geq 0. && u \in U \end{aligned} \tag{19}$$

In case of $T_{P,\varphi}$, optimization problem (18) obtains the form

$$\begin{aligned} \text{minimize} \quad & p \sum_{u \in U} x_u + (1-p) \sum_{u \in U} y_u, \\ \text{subject to} \quad & \alpha_v \widetilde{R}_\varphi(u, v) \leq \alpha_u, && u, v \in U \\ & x_u - y_u = \bar{A}_\varphi(u) - \alpha_u, && u \in U \\ & 0 \leq \alpha_u \leq 1,\ x_u \geq 0,\ y_u \geq 0. && u \in U \end{aligned} \tag{20}$$

Summarizing, for quantile risk and a continuous Archimedean t-norm, the optimization problem (18) 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 5.1. *Constraints $0 \leq \alpha_u \leq 1, u \in U$ in (19) and (20), 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 (19) and (20) 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 a 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 2.2 does not hold anymore and using $L_p(\bar{A}_\varphi(u), \alpha_u)$ 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

$$\begin{aligned} \text{minimize} \quad & \sum_{u \in U} (\alpha_u - \bar{A}_\varphi(u))^2, \\ \text{subject to} \quad & \alpha_u - \alpha_v + 1 \geq \widetilde{R}_\varphi(u, v), && u, v \in U \\ & 0 \leq \alpha_u \leq 1, && u \in U \end{aligned} \tag{21}$$

while for $T_{P,\varphi}$ the corresponding problem is

$$\begin{aligned} \text{minimize} \quad & \sum_{u \in U} (\alpha_u - \bar{A}_\varphi(u))^2, \\ \text{subject to} \quad & \alpha_v \widetilde{R}_\varphi(u, v) \leq \alpha_u, && u, v \in U \\ & 0 \leq \alpha_u \leq 1. && u \in U \end{aligned} \tag{22}$$

Using a similar argument as in Proposition 5.1, 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 the appendix of this paper. Namely, the dual versions of problems (19) and (20) 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 Appendix A while we show how to model dual problems of (19) and (20) as the min-cost flow problem and a variation of the min-cost flow problem, respectively, in Appendix B. In the same section, we provide a greedy algorithm to solve the aforementioned

Table 8
Granular approximations in the classification case for the p-quantile loss and T-equivalence relation.

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 9
Granular approximations in the classification case for the p-quantile loss and T-preorder relation.

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 10
Granular approximations in the classification case for the squared error loss and T-equivalence relation.

instance	1	2	3	4	5	6
degree	0.965	0.817	0.517	0.517	0.053	0.130

Table 11
Granular approximations in the classification case for the squared error loss and T-preorder relation.

instance	1	2	3	4	5	6
degree	0.960	0.607	0.607	0.607	0.217	0.000078

variation based on the algorithm that solves the original min-cost flow problem. Since the algorithm is new, we provide its proof of correctness in Appendix C. The combinatorial approach or duality of the quadratic programs was not discussed.

The geometrical approach includes the aforementioned simplex methods. They are based on geometrical structures that are created in space by constraints and the objective function. There are many softwares that are able to solve linear and quadratic programs like Gurobi [46] and Mosek [47]. We need to bear in mind that the proposed optimization problems have $O(|U|)$ variables and $O(|U|^2)$ constraints which lead to the constraint matrix with $O(|U|^3)$ entries. For a large sample size, dealing with such a matrix can be computationally demanding. However, the matrix is sparse (a vast majority of entries are 0) and our internal experiments showed that the Mosek solver can be used as an efficient option to deal with this sparse constraint matrix.

Example 5.1. This example continues with the data introduced in Section 3. We want to calculate granular approximations using optimization procedures (19) and (21) that are developed for the Łukasiewicz t-norm T_L.

First, we calculate the granular approximation of the classification dataset from Table 2 using T_L-equivalence relation (14) and quantile loss L_p. The relation matrix from Table 3 is passed together with the decision attribute to the optimization problem (19) with probability parameters $p \in \{0, 0.25, 0.5, 0.75, 1\}$. The obtained granular approximations are given in Table 8.

In Table 9, we present the calculated granular approximations using T_L-preorder relation (13) while the remaining parameters are the same as in Table 8.

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 8, 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 9, the degree is not greater than 0.390.

The granular approximations obtained using optimization problem (21) and T_L-equivalence relation (14) are shown in Table 10, while the output of the same optimization procedure using T_L-preorder relation (13) is provided in Table 11.

Table 12
Granular approximations in the regression case for the p-quantile loss and T-equivalence relation.

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 13
Granular approximations in the regression case for the p-quantile loss and T-preorder relation.

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 14
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	0.343	0.718	0.729	0.400	0.421

Table 15
Granular approximations in the classification case for the squared error loss and T-preorder relation.

instance	1	2	3	4	5	6
degree	0.770	0.477	0.820	0.560	0.400	0.352

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, where we identified that instances $u \equiv 6$ and $v \equiv 2$ were inconsistent, using results from Table 10, we obtain $T_L(\widetilde{R}(u,v)\hat{A}(v)) = T_L(0.312, 0.817) = 0.129 \leq 0.13 = \hat{A}(u)$, i.e., they are now consistent. If we use the results from Table 11, we have that $T_L(0.312, 0.607) = 0 \leq 0.000078 = \hat{A}(u)$, i.e., they are consistent. The values of the fuzzy relations in these examples are obtained from Tables 3 and 4.

We perform the same calculations for the regression data from Section 3 provided in Table 5. In order to compute the granular approximations w.r.t. quantile loss and T_L-equivalence relation (14), we pass the relation values from Table 6 and the decision attribute from Table 5 to optimization procedure (19) with probability parameters $p \in \{0, 0.25, 0.5, 0.75, 1\}$. The obtained granular approximations are given in Table 12.

In Table 13 we calculate the granular approximations using T_L-preorder relation (13) while the other parameters are the same as in Table 12.

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 5. We interpret the values in a way that, for example, in the third row of Table 12 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 13 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 (21) are shown in Tables 14 and 15 for T_L-equivalence (14) and T_L-preorder (13).

In the case of Table 14, we say that the expected expensiveness of instance 4 is equal to 0.729, while in the case of Table 15 the expected expensiveness of instance 4 is equal to 0.56.

We again continue the example from Section 3 where we identified that instances $u \equiv 2$ and $v \equiv 3$ are inconsistent. Using estimated values from Table 14 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 15 we have that $T(\widetilde{R}(u,v)\hat{A}(v)) = T(0.625, 0.82) = 0.445 \leq 0.477 = \hat{A}(u)$, 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. Properties

In this section, we prove some properties of the granular approximations obtained in Section 5. The first two propositions show that the proposed approach is indeed a generalization of both the KS approach [12] for the binary classification case, and of the standard fuzzy rough set approximations.

Proposition 6.1. *If $\widetilde{R}$ and $\bar{A}$ are crisp, then Problem (18) is reduced to Problem (6) for $K = 2$.*

Proof. If $\bar{A}$ is crisp, it is obvious that the objective function from (18) corresponds to the objective function from (6) for $K = 2$, where the labels with value 1 are those that are more preferred. Regarding the constraints, we examine the consistency conditions in the 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 (10). 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 (6). □

Proposition 6.2. *The respective lower fuzzy rough approximations are solutions of the optimization problems (19) and (20) 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 (19) and (20) are considered in terms of α and not in terms of $\hat{A}$, they can be seen as problem (18) 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 to 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 2.5, 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

$$\begin{aligned}\hat{A}^*(u) &= \varphi^{-1}(\alpha_u^*)\\ &= \varphi^{-1}(\max_{v \in U} T_L(\widetilde{R}_\varphi(v,u), \bar{A}_\varphi(v)))\\ &= \max_{v \in U} \varphi^{-1}(T_L(\varphi(\widetilde{R}(v,u)), \varphi(\bar{A}(v))))\\ &= \max_{v \in U} T_{L,\varphi}(\widetilde{R}(v,u), \bar{A}(v)) = \overline{\text{apr}}_{\widetilde{R}}^{\max, T_{L,\varphi}}(A)(u).\end{aligned}$$

The derivation for T_P is the same.

The proof for the lower approximation is analogous. □

We examine Proposition 6.2 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 [48]. 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 extreme since the resulting approximations are either a subset (lower approximation) or a superset (upper approximation) of the original (fuzzy) set. It is thus 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. *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 (19) or (20) 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 Appendix D. It relies on the greedy combinatorial approach presented in the previous sections of the Appendix, hence those previous sections are necessary for the understanding of the proof. □

In Proposition 6.3, 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 6.2. 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 2.3. The following lemma and proposition extend that property to granularly representable sets and granular approximations. The duality property is particularly important for binary classification problems. It ensures that granular approximations of two different decision classes are complementary w.r.t. a given fuzzy negation N.

Lemma 6.1. *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, 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 (2.4) while the second is the residuation property. □

In the proof of Lemma 6.1, the implication becomes an equivalence if we use IMTL triplets as operators.

Proposition 6.4. *Let $\alpha_u, u \in U$ be a minimizer of the optimization problem (18) 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 (18) 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 6.4 where the only difference is that the loss function stays the same in the dual problems.

7. Conclusion and future work

In this paper, we introduced a novel statistical learning approach for handling inconsistencies in classification and regression problems with respect to a fuzzy relation. Our work was motivated by the method introduced by Kotłowski and Słowiński [12] for handling monotone inconsistency 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.

The novel approach produces a granular approximation of a fuzzy set. The approximation is granularly representable (without inconsistencies) and its difference from the original fuzzy set is minimal (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 their important properties. We also presented two didactic examples; one for a binary classification problem and another 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.

We wish to stress that our contribution is in the theoretical development of granular approximations that can be applied in many different tasks of machine learning, motivated by the applications of the rough sets. Therefore, any experimental evaluation of the granular approximations will depend on the chosen application and the associated data. As was already mentioned in the introduction, the possible applications are in fuzzy rough set-based methods and fuzzy rule induction, so our future work will mainly go in this direction.

CRediT authorship contribution statement

Marko Palangetić: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Chris Cornelis:** Resources, Supervision, Writing – review & editing. **Salvatore Greco:** Resources, Supervision. **Roman Słowiński:** Resources, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

Marko Palangetić and Chris Cornelis would like to thank Odysseus project from Flanders Research Foundation (FWO), grant no. G0H9118N, for funding their research. Salvatore Greco wishes to acknowledge the support of the Ministero dell'Istruzione, dell'Universitá e della Ricerca (MIUR) - PRIN 1576 2017, project "Multiple Criteria Decision Analysis and Multiple Criteria Decision Theory", grant 2017CY2NCA. Roman Słowiński is acknowledging the support of Polish Ministry of Education and Science, grant 0311/SBAD/0726.

Appendix A. Minimum-cost flow problem

This section is based on the monograph [49], 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 satisfies 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.

Algorithm 1 Successive Shortest Paths.

Input: Flow network $F = (G, E)$.
Output: Flow z.
Set initial flow $z_{i,j} = 0$, $(i,j) \in E$
Set initial residual network $F' = F$
while there exist supply/demand values different from 0 **do**
    Pick supply node i and demand node j
    Calculate the shortest path P from i to j using cost values from F'
    Send the largest possible amount of flow through P
    Update F'
Reconstruct z from F'

Formally, we have the following problem:

$$\text{minimize} \quad \sum_{(i,j)\in E} c_{i,j} z_{i,j}, \tag{A.1a}$$

$$\text{subject to} \quad \sum_{j:(i,j)\in E} z_{i,j} - \sum_{j:(j,i)\in E} z_{j,i} = b_i, \qquad i \in G \tag{A.1b}$$

$$0 \le z_{i,j} \le l_{i,j}. \qquad (i,j) \in E \tag{A.1c}$$

We distinguish two sets of constraints in the previous optimization problem: balance constraints (A.1b) and capacity constraints (A.1c). 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 (A.1), 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 (A.1). 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 A.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 A.1, a simple algorithm can be constructed to solve (A.1). 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.

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 [49]. 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

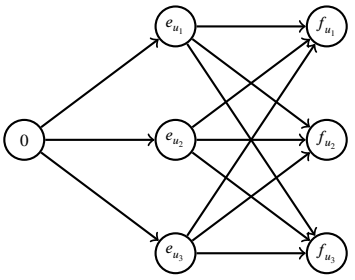

Fig. B.2. Flow modeled as a bipartite graph.

$$\begin{aligned} &\text{minimize} && \sum_{(i,j)\in E} c_{i,j} z_{i,j}, \\ &\text{subject to} && \sum_{j:(i,j)\in E} m_{i,j} z_{i,j} - \sum_{j:(j,i)\in E} z_{j,i} = b_i, && i \in G \\ & && 0 \le z_{i,j} \le l_{i,j}, && (i,j) \in E. \end{aligned} \tag{A.2}$$

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 (A.2). Fortunately, our particular case of (A.2) allows obtaining similar properties as in the ordinary minimum-cost flow problem (A.1).

Appendix B. Duality and the combinatorial approach

In this section, the dual optimization problems of (19) and (20) 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 (19). 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

$$\begin{aligned} &\text{maximize} && p\sum_{u\in U} \alpha_u - \sum_{u\in U} y_u, \\ &\text{subject to} && \alpha_v - \alpha_u \le M(u,v), && u,v \in U \\ & && \alpha_u - y_u \le \bar{A}_\varphi(u), && u \in U \\ & && y_u \ge 0 && u \in U. \end{aligned} \tag{B.1}$$

Its dual problem is then

$$\begin{aligned} &\text{minimize} && \sum_{u,v\in U} M(u,v) z_{u,v} + \sum_{u\in U} \bar{A}_\varphi(u) z_{0,u} \\ &\text{subject to} && -z_{0,u} + \sum_{v\in U} z_{u,v} - \sum_{v\in U} z_{v,u} = -p, && u \in U \\ & && z_{0,u} \le 1. && u \in U. \end{aligned} \tag{B.2}$$

In (B.2), variables $z_{u,v}, u,v \in U$, correspond to the first set of constraints from primal (B.1), while variables $z_{0,u}, u \in U$, correspond to the second set of constraints from the primal. The first set of constraints in (B.2) 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 (B.2) 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 $M(u,v) + M(v,w) \ge M(u,w)$ for $u,v,w \in U$. Using this fact, we have that there is an optimal flow that 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.

In Fig. B.2, 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 Appendix A.

The bipartite network representation is useful from the perspective of 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 (20).

The next question is how to reconstruct the optimal solution of the primal problem, i.e., to calculate α^* from a solution of the dual z^*. Following the duality theory provided in [49], 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 (20). The linear program here can be rewritten similarly as (B.1), just with 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

$$\begin{aligned} \text{minimize} \quad & \sum_{u \in U} \bar{A}_\varphi(u) z_{0,u}, \\ \text{subject to} \quad & -z_{0,u} + \sum_{v \in U} z_{u,v} - \sum_{v \in U} \widetilde{R}_\varphi(v, u) z_{v,u} = -p, \quad u \in U \\ & z_{0,u} \leq 1. \qquad\qquad\qquad\qquad\qquad\quad u \in U \end{aligned} \tag{B.3}$$

The difference between (B.2) and (B.3) 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 (B.2), 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 Fig. B.2. 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 a 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.

To calculate the smallest possible cost from the supply node, we can use the 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 the least costly way to transport the flow.

Algorithm 2 Generalized successive shortest paths.

Input: Bipartite flow network F.
Output: Flow z.
Set initial flow $z_{i,j} = 0$, $(i,j) \in E$
Set initial residual network $F' = F$
while there exists a demand value different from 0 **do**
    Pick a demand node i
    Calculate the smallest possible cost from the supply node to i
    Calculate the largest amount of flow that can be sent through the least costly path
    Send the calculated flow through the least costly path
    Update F'
Reconstruct z' from F'

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 $P = (0, e_{u_1}, f_{u_2}, e_{u_3}, \dots, f_{u_k})$ 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 at 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 Appendix C, 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 Appendix C, 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.

Appendix C. Proof of correctness for Algorithm 2

In this section 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 C.1. *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 (B.3) with a positive constant C, the optimal solution is Cz^* where z^* is the solution of the initial problem. For some parameter a in (B.3) 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 (B.3). If we multiply the right side of the constraints in (B.3) 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 Fig. C.3.

In Fig. C.3, 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 $J(e_{u_1}, \dots, e_{u_2})$ 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 $A_\varphi(u_1) < J(e_{u_1}, \dots, e_{u_2}) A_\varphi(u_2)$. 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

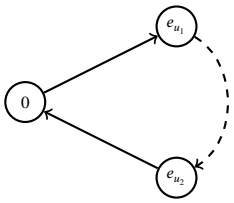

Fig. C.3. Cycle in a generalized bipartite network.

flow along it, the cost of that flow is $A_\varphi(u_1) - J(e_{u_1}, \dots, e_{u_2})A_\varphi(u_2)$, 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 C.2. *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 C.3. *Solution z^* is optimal in the generalized bipartite network if and only if its residual network does not contain negative cost cycles.*

Proof. (⇒) 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.

(⇐) 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 $z'_{0,e_{u_1}} \le 1$ then $z''_{0,e_{u_1}} \le 1 - z^*_{0,e_{u_1}}$ 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 $z'_{0,e_{u_1}} \ge 0$ then $z''_{e_{u_1},0} \le z^*_{0,e_{u_1}}$ 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 Fig. C.3, i.e., it has a cycle representation. Proposition C.2 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 $b_1 \ge b_2$. 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'', the 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 C.4. *Algorithm 2 returns an optimal solution.*

Proof. Assume that in one iteration of Algorithm (2), the shortest path had the form $P_1 = (0, e_{u_2}, \dots, f_{u_3})$ and that after the step, the negative cost cycle $(0, e_{u_1}, \dots, f_{u_3}, \dots, e_{u_2}, 0)$ was formed. The negative cost cycle is formed from the path $P_2 = (0, e_{u_1}, \dots, f_{u_3})$ and the reverse path P_1. The model of such a cycle is represented in Fig. C.4. The dots in the cycle as well as the 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

$$A_\varphi(u_1) < J(e_{u_1}, \dots, f_{u_3})J(f_{u_3}, \dots, e_{u_2})A_\varphi(u_2).$$

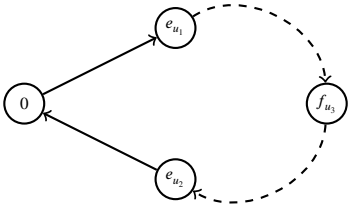

Fig. C.4. Cycle after one step of Algorithm 2.

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 C.3. □

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 C.5. *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 linear programs.

According to the strong duality theorem [50], 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 (B.1) and in (B.3) are equal, i.e.,

$$\sum_{u\in U} \bar{A}_\varphi(u) z^*_{0,u} = \sum_{u\in U} p\alpha^*_u - \sum_{u\in U} \max(\alpha^*_u - \bar{A}_\varphi(u), 0). \tag{C.1}$$

In the previous expression, y_u is replaced with its definition. In an optimal solution, for $u \in U$, we have that

$$\sum_{v\in U} z^*_{u,v} = z^*_{0,u}, \quad \sum_{v\in U} \widetilde{R}_\varphi(v,u) z^*_{v,u} = p. \tag{C.2}$$

We have the following equalities:

$$\begin{aligned}
\sum_{u\in U} \max(\alpha^*_u - \bar{A}_\varphi(u), 0) &= \sum_{u\in U} p\alpha^*_u - \sum_{u\in U} \bar{A}_\varphi(u) z^*_{0,u} \\
&= \sum_{u\in U} p\alpha^*_u - \sum_{u\in U} (\bar{A}_\varphi(u) - \alpha^*_u) z^*_{0,u} - \sum_{u\in U} \alpha^*_u z^*_{0,u} \\
&= \sum_{u\in U} p\alpha^*_u - \sum_{u\in U} (\bar{A}_\varphi(u) - \alpha^*_u) z^*_{0,u} - \sum_{u\in U} \alpha^*_u \sum_{v\in U} z^*_{u,v} \\
&= \sum_{u\in U} p\alpha^*_u - \sum_{u\in U} (\bar{A}_\varphi(u) - \alpha^*_u) z^*_{0,u} \\
&\quad - \sum_{u,v\in U} (\alpha^*_u - \widetilde{R}_\varphi(u,v)\alpha^*_v) z^*_{u,v} - \sum_{u,v\in U} \widetilde{R}_\varphi(u,v)\alpha^*_v z^*_{u,v} \\
&= \sum_{u\in U} p\alpha^*_u - \sum_{u\in U} (\bar{A}_\varphi(u) - \alpha^*_u) z^*_{0,u}
\end{aligned}$$

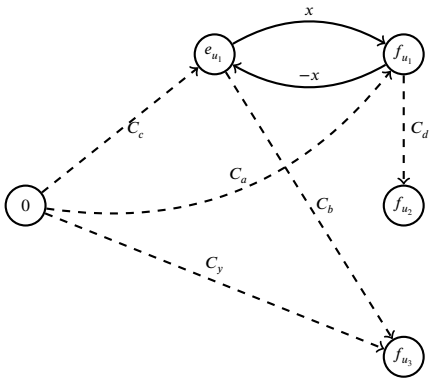

Fig. D.5. Flow modeled as a bipartite graph.

$$
\begin{aligned}
&- \sum_{u,v\in U} (\alpha_u^* - \widetilde{R}_\varphi(u,v)\alpha_v^*) z_{u,v}^* - \sum_{v\in U} \alpha_v^* \sum_{u\in U} \widetilde{R}_\varphi(u,v) z_{u,v}^* \\
&= \sum_{u\in U} (\alpha_u^* - \bar{A}_\varphi(u)) z_{0,u}^* - \sum_{u,v\in U} (\alpha_u^* - \widetilde{R}_\varphi(u,v)\alpha_v^*) z_{u,v}^*.
\end{aligned}
$$

The second equality holds because of the left expression in (C.2) while the last equality holds because the right expression in (C.2). 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 C.5.

Appendix D. Proof of Proposition 6.3

Let $\alpha_u^p = \varphi(\hat{A}_p(u))$ and $\alpha_u^q = \varphi(\hat{A}_q(u))$ for $u \in U$. Then $\hat{A}_p(u) \leq \hat{A}_q(u) \Leftrightarrow \alpha_u^p \leq \alpha_u^q$. 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 Fig. D.5.

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 $(0,\dots,e_{u_1},f_{u_1},\dots,f_{u_2})$ 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 $(0,\dots,f_{u_3})$ with cost C_y while after the previous step and after adding reverse edge (f_{u_1},e_{u_1}) the shortest path is $(0,\dots,f_{u_1},e_{u_1},\dots,f_{u_3})$ 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 $(0,\dots,e_{u_1},f_{u_1},\dots,f_{u_2})$, 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.

References

[1] P.A. Gutiérrez, M. Perez-Ortiz, J. Sanchez-Monedero, F. Fernandez-Navarro, C. Hervas-Martinez, Ordinal regression methods: survey and experimental study, IEEE Trans. Knowl. Data Eng. 28 (1) (2015) 127–146.
[2] P. Bellmann, F. Schwenker, Ordinal classification: working definition and detection of ordinal structures, IEEE Access 8 (2020) 164380–164391.
[3] R. Chandrasekaran, Y.U. Ryu, V.S. Jacob, S. Hong, Isotonic separation, INFORMS J. Comput. 17 (4) (2005) 462–474.
[4] S. Greco, B. Matarazzo, R. Słowiński, A new rough set approach to evaluation of bankruptcy risk, in: Operational Tools in the Management of Financial Risks, Springer, 1998, pp. 121–136.
[5] R. Potharst, A.J. Feelders, Classification trees for problems with monotonicity constraints, ACM SIGKDD Explor. Newsl. 4 (1) (2002) 1–10.
[6] J.-R. Cano, P.A. Gutiérrez, B. Krawczyk, M. Woźniak, S. García, Monotonic classification: an overview on algorithms, performance measures and data sets, Neurocomputing 341 (2019) 168–182.
[7] S. González, S. García, S.-T. Li, R. John, F. Herrera, Fuzzy k-nearest neighbors with monotonicity constraints: moving towards the robustness of monotonic noise, Neurocomputing 439 (2021) 106–121.
[8] A. Campagner, D. Ciucci, E. Hüllermeier, Rough set-based feature selection for weakly labeled data, Int. J. Approx. Reason. 136 (2021) 150–167.
[9] Z. Pawlak, Rough sets, Int. J. Comput. Inf. Sci. 11 (5) (1982) 341–356.
[10] S. Greco, B. Matarazzo, R. Słowiński, Rough sets theory for multicriteria decision analysis, Eur. J. Oper. Res. 129 (1) (2001) 1–47.
[11] M. Palangetić, C. Cornelis, S. Greco, R. Słowiński, Granular representation of OWA-based fuzzy rough sets, Fuzzy Sets Syst. (2021).
[12] W. Kotłowski, R. Słowiński, Statistical approach to ordinal classification with monotonicity constraints, in: Preference Learning ECML/PKDD 2008 Workshop, 2008.
[13] W. Kotłowski, K. Dembczyński, S. Greco, R. Słowiński, Stochastic dominance-based rough set model for ordinal classification, Inf. Sci. 178 (21) (2008) 4019–4037.
[14] K. Dembczyński, W. Kotłowski, S. Greco, R. Słowiński, Ensemble of decision rules for ordinal classification with monotonicity constraints, in: G. Wang, et al. (Eds.), Rough Sets and Knowledge Technology (RSKT 2008), in: LNAI, vol. 5009, Springer, Berlin, 2008, pp. 260–267.
[15] R.E. Barlow, H.D. Brunk, The isotonic regression problem and its dual, J. Am. Stat. Assoc. 67 (337) (1972) 140–147.
[16] D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets, Int. J. Gen. Syst. 17 (2–3) (1990) 191–209.
[17] S. Greco, B. Matarazzo, R. Słowiński, Fuzzy extension of the rough set approach to multicriteria and multiattribute sorting, in: Preferences and Decisions Under Incomplete Knowledge, Springer, 2000, pp. 131–151.
[18] L.A. Zadeh, Fuzzy sets and information granularity, Adv. Fuzzy Set Theory Appl. 11 (1979) 3–18.
[19] L.A. Zadeh, Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Sets Syst. 90 (2) (1997) 111–127.
[20] T.Y. Lin, et al., Granular computing on binary relations i: data mining and neighborhood systems, granular computing on binary relations ii: rough set representations and belief functions, in: Rough Sets in Knowledge Discovery, vol. 1, 1998, pp. 107–140.
[21] Y. Yao, Granular computing using neighborhood systems, in: Advances in Soft Computing, Springer, 1999, pp. 539–553.
[22] Z. Pawlak, Granularity of Knowledge, Indiscernibility and Rough Sets, 1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36228), vol. 1, IEEE, 1998, pp. 106–110.
[23] Z. Pawlak, Granularity, multi-valued logic, Bayes' theorem and rough sets, in: Data Mining, Rough Sets and Granular Computing, Springer, 2002, pp. 487–498.
[24] Y. Yao, Rough Sets, Neighborhood Systems and Granular Computing, Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No. 99TH8411), vol. 3, IEEE, 1999, pp. 1553–1558.
[25] S. Zhao, E.C. Tsang, D. Chen, X. Wang, Building a rule-based classifier—a fuzzy-rough set approach, IEEE Trans. Knowl. Data Eng. 22 (5) (2009) 624–638.
[26] S. Zhao, Z. Dai, X. Wang, P. Ni, H. Luo, H. Chen, C. Li, An accelerator for rule induction in fuzzy rough theory, IEEE Trans. Fuzzy Syst. 29 (12) (2021) 3635–3649.
[27] R. Jensen, C. Cornelis, Fuzzy-rough nearest neighbour classification and prediction, Theor. Comput. Sci. 412 (42) (2011) 5871–5884.
[28] B. Behera, G. Kumaravelan, Text document classification using fuzzy rough set based on robust nearest neighbor (frs-rnn), Soft Comput. 25 (15) (2021) 9915–9923.
[29] Z. Liu, S. Pan, Fuzzy-rough instance selection combined with effective classifiers in credit scoring, Neural Process. Lett. 47 (1) (2018) 193–202.
[30] C. Wang, Y. Huang, W. Ding, Z. Cao, Attribute reduction with fuzzy rough self-information measures, Inf. Sci. 549 (2021) 68–86.
[31] W. Gilchrist, Statistical Modelling with Quantile Functions, CRC Press, 2000.
[32] M. Shaked, J.G. Shanthikumar, Stochastic Orders, Springer Science & Business Media, 2007.
[33] J.O. Berger, Statistical Decision Theory and Bayesian Analysis, Springer Science & Business Media, 2013.
[34] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Courier Corporation, 1998.
[35] E.P. Klement, R. Mesiar, E. Pap, Triangular Norms, vol. 8, Springer Science & Business Media, 2013.
[36] M. Palangetić, C. Cornelis, S. Greco, R. Słowiński, Fuzzy extensions of the dominance-based rough set approach, Int. J. Approx. Reason. 129 (2021) 1–19.
[37] L. Yang, A. Shami, On hyperparameter optimization of machine learning algorithms: theory and practice, Neurocomputing 415 (2020) 295–316.
[38] R.I. John, P.R. Innocent, Modeling uncertainty in clinical diagnosis using fuzzy logic, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 35 (6) (2005) 1340–1350.
[39] I. Özkan, I.B. Türkşen, Uncertainty and fuzzy decisions, in: Chaos Theory in Politics, Springer, 2014, pp. 17–27.
[40] T.J. Davis, C.P. Keller, Modelling uncertainty in natural resource analysis using fuzzy sets and Monte Carlo simulation: slope stability prediction, Int. J. Geogr. Inf. Sci. 11 (5) (1997) 409–434.
[41] R.R. Yager, Set-based representations of conjunctive and disjunctive knowledge, Inf. Sci. 41 (1) (1987) 1–22.
[42] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets Syst. 1 (1) (1978) 3–28.
[43] M.L. Puri, D.A. Ralescu, Fuzzy random variables, J. Math. Anal. Appl. 114 (2) (1986) 409–422, https://doi.org/10.1016/0022-247X(86)90093-4.
[44] W. Rudin, Real and Complex Analysis, McGraw-Hill Book Company, 1987.
[45] R. Koenker, K.F. Hallock, Quantile regression, J. Econ. Perspect. 15 (4) (2001) 143–156.
[46] L.L.C. Gurobi Optimization, Gurobi Optimizer Reference Manual, https://www.gurobi.com, 2022.
[47] M. ApS, The MOSEK optimization toolbox for MATLAB manual, Version 9.0, http://docs.mosek.com/9.0/toolbox/index.html, 2019.

[48] M. Palangetić, C. Cornelis, S. Greco, R. Słowiński, Rough sets meet statistics - a new view on rough set reasoning about numerical data, in: International Joint Conference on Rough Sets, Springer, 2020, pp. 78–92.
[49] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows, Massachusetts Institute of Technology, 1988.
[50] J. Matousek, B. Gärtner, Understanding and Using Linear Programming, Springer Science & Business Media, 2007.