# Relation-based fuzzy granular approximation

Marko Palangetić[1 [0000−0002−6366−0634]], Chris Cornelis[1 [0000−0002−7854−6025]], Salvatore Greco[2,3 [0000−0001−8293−8227]], and Roman Słowiński[4,5 [0000−0002−5200−7795]]

[1] Ghent University, Ghent, Belgium
{marko.palangetic, chris.cornelis}@ugent.be
[2] University of Catania, Catania, Italy
[3] University of Portsmouth, Portsmouth, United Kingdom
salgreco@unict.it
[4] Poznań University of Technology, Poznań, Poland
[5] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
roman.slowinski@cs.put.poznan.pl

**Abstract.** We introduce a formal definition of the relation-based fuzzy granular approximations (RFGA). These approximations may be seen as a generalization of the fuzzy rough approximations (FRA). The new definition yields a family of fuzzy sets that extends the family of lower and upper fuzzy approximations, and inherits their granular properties. Moreover, a fuzzy relation is used to represent the decision attribute instead of a fuzzy set, like in the FRA. Besides the formal definition, we provide a way of calculating the RFGA in specific cases, and we discuss possible applications of the RFGA in Machine Learning.

**Keywords:** Fuzzy rough sets · Machine Learning · Granular approximations.

## 1 Introduction

Rough sets were introduced by Pawlak [13] to deal with inconsistencies within information tables where objects are described by a set of attributes. Pawlak's approach produces two sets, called lower and upper approximation, which represent objects being, respectively, necessarily consistent (lower approximation), and possibly consistent (upper approximation) with knowledge contained in the information table. The original theory was designed to deal with nominal attributes, and relies on an equivalence relation, expressing indiscernibility between objects. Pawlak's original theory was later named the Indiscernibility-based Rough Set Approach (IRSA).

On the other hand, fuzzy set theory [17] studies the gradual truth of logical statements, and is used extensively in modeling imprecise and vague information. The combination of fuzzy sets and IRSA was first proposed by Dubois and Prade [4], allowing to approximate fuzzy sets using a fuzzy $T$-equivalence relation. A similar extension of Dominance-based Rough Set Approach (DRSA) to fuzzy set theory was proposed by Greco et al. [5].

Rough sets are very useful from the point of view of granular computing, as they possess a so-called granular representation; indeed, lower and upper approximations can be represented as unions of simple sets or granules, induced from the data [16]. In contrast to crisp sets, the granular properties of fuzzy rough sets do not stem directly from the proposed definitions. Degang et al. [3] were the first to show that fuzzy IRSA has indeed a granular representation which means that fuzzy rough approximations can be represented as a union of simple fuzzy sets or fuzzy granules.

The granular representation of rough sets and fuzzy rough sets is, in particular, very useful from the perspective of rule induction. The problem of rule induction for classification tasks amounts to generating a set of rules which relate description of objects by subsets of attributes with particular decision classes. Basic granules, from which rough sets are composed, can be interpreted as human readable "*if..., then...*" rules, and can be used to construct a rule-based inference system as a prediction model. A well-known examples of rule induction algorithms are the LEM2 algorithm [6] and the MODLEM algorithm [15]. Similarly, the granularity of fuzzy rough sets has also been used for rule induction. In this case, we obtain a fuzzy inference system, with flexible fuzzy rules instead of strict crisp rules [9, 18]. The main advantage of fuzzy rules is that they can model complex shapes of data, and still keep an intuitive interpretation of these shapes.

Palangetić et al. [12] introduced the concept of granularly representable sets as an initial step towards the definition of granular approximations. Granular approximations may be seen as generalizations of rough sets (in both crisp and fuzzy case), i.e., they are granularly representable sets which are neither lower nor upper approximations, but lie between these two extreme sets. In this paper, we introduce a formal definition of such granular approximation in the fuzzy case, called Relation-based Fuzzy Granular Approximation (RFGA). The novelty of the current proposal is that instead of approximating fuzzy sets (like in the definition of the lower and upper fuzzy rough approximations) we approximate a fuzzy relation. Such fuzzy relation measures the similarity of objects on the decision attribute. The motivation to use a fuzzy relation may be found in Data analysis and Machine Learning. Fuzzy relations are suitable for modeling different types of data and may be used on many types of Machine Learning problems (classification, regression ...).

The remainder of this paper is structured as follows. In Section 2, we recall some useful preliminaries about fuzzy, rough and granularly representable sets. Section 3 deals with the derivation of the relation-based fuzzy granular representation. Section 4 explains how to solve the system of equations that is raised in the definition of RFGA from Section 3. Sections 5 discusses the use of the proposed approach in Machine Learning, while Section 6 provides initial experimental results of an application to real data. Section 7 contains our conclusions and outlines future work.

## 2 Preliminaries

### 2.1 Fuzzy logic connectives

In this subsection, the definitions and terminology are based on [10]. 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.

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

Table 1: Some common $t$-norms and their $R$-implicators

We call a $t$-norm Archimedean if

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

$T_P$, $T_L$ and $T_D$ are Archimedean, while $T_M$ and $T_{nM}$ are not.

A decreasing generator is a continuous decreasing function $f : [0,1] \to \overline{\mathbb{R}}^+$ such that $f(1) = 0$. The following proposition characterizes continuous Archimedean $t$-norms.

**Proposition 1.** *A mapping $T : [0,1]^2 \to [0,1]$ is a continuous, Archimedean t-norm if and only if there exists a decreasing generator $f$ such that for $(x,y) \in [0,1]^2$:*

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

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

$$T(x,y) \le z \Leftrightarrow x \le I(y,z).$$

It is well-known that the residuation property holds 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 right hand side 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.

If $T$ is a continuous Archimedean $t$-norm with a decreasing generator $f$, then its R-implicator has the following form:

$$I(x, y) = f^{-1}(\max(f(y) - f(x), 0)).$$

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 an implicator $I$, we define the negator induced by $I$ as $N(x) = I(x, 0)$. If $T$ is a continuous Archimedean $t$-norm, then the induced negator of $I_T$ has the form

$$N(x) = f^{-1}(f(0) - f(x)).$$

Such negator is involutive if and only if $f(0) < \infty$. The standard negator is obtained by taking the decreasing generator of the Łukasiewicz $t$-norm $f(x) = 1 - x$. A $t$-norm for which the induced negator of its R-implicator is involutive is called an IMTL $t$-norm. For an implicator $I$ and its induced negator $N$ it holds that

$$I(x, y) \leq I(N(y), N(x)), \tag{1}$$

where the equality holds if $N$ is involutive.

### 2.2 Fuzzy sets and fuzzy relations

Given a non-empty set $U$, a fuzzy set $A$ on $U$ 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 $\{0, 1\}$ then 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 $\alpha$-level set of fuzzy set $A$ is a crisp set defined as $A_\alpha = \{u \in U; A(u) \geq \alpha\}$.

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

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

A reflexive, symmetric and $T$-transitive fuzzy relation is called $T$-equivalence relation.

An example of a $T_L$-equivalence relation is the triangular similarity which will be used later. For a conditional attribute $q$, it is defined as

$$R_q(u,v) = \max\left(1 - \frac{|f(u,q) - f(v,q)|}{range(q)}, 0\right) \tag{2}$$

while the overall relation is then $R(u,v) = \min_q R_q(u,v)$. More details on such similarity relation are provided in [11].

### 2.3 Fuzzy rough and granular approximations

This subsection is based on [12]. Let $R$ be a $T$-equivalence relation. The well known fuzzy lower and upper approximations are defined as:

$$\begin{aligned} \underline{\text{apr}}_R^{\min,I}(A)(u) &= \min(I(R(u,v), A(v)); v \in U) \\ \overline{\text{apr}}_R^{\max,T}(A)(u) &= \max(T(R(u,v), A(v)); v \in U). \end{aligned} \tag{3}$$

A fuzzy granule is defined as a parametric fuzzy set

$$R_\beta(u) = \{(v, T(R(u,v), \beta); v \in U\}. \tag{4}$$

A fuzzy set $A$ is granularly representable (GR) if

$$A = \bigcup \{R_{A(u)}(u); u \in U\},$$

where the union is defined with max operator. It holds that $\underline{\text{apr}}_R^{\min,I}(A)(u)$ is the largest GR set contained in $A$, while $\overline{\text{apr}}_R^{\max,T}(A)(u)$ is the smallest GR set containing $A$.

## 3 Definition of RFGA

In this section we provide a formal definition of the relation-based fuzzy granular approximation (RFGA). Note that from now on we assume that $R$ is a $T$-equivalence fuzzy relation for some left-continuous $t$-norm $T$. We start with some basic propositions and definitions.

**Definition 1.** *Two fuzzy sets $A$ and $B$ are called $T$-disjoint if*

$$T(A(u), B(u)) = 0 \text{ for every } u \in U.$$

For the fuzzy granules we have the following property:

**Proposition 2.** *Two fuzzy granules $R_{\beta_1}(u)$ and $R_{\beta_2}(v)$ are $T$-disjoint if and only if*

$$T(\beta_1, \beta_2) \leq N(R(u,v)). \tag{5}$$

*Proof.* The statement that two granules are $T$-disjoint is equivalent to:

$$\begin{aligned}
&\max_{w\in U} T(T(R(u,w),\beta_1),T(R(w,v),\beta_2)) = 0\\
\Leftrightarrow &\max_{w\in U} T(T(R(u,w),R(w,v)),T(\beta_1,\beta_2)) = 0\\
\Leftrightarrow &T\left(\max_{w\in U} T(R(u,w),R(w,v)),T(\beta_1,\beta_2)\right) = 0\\
\Leftrightarrow &T(R(u,v),T(\beta_1,\beta_2)) = 0\\
\Leftrightarrow &T(\beta_1,\beta_2) \leq I(R(u,v),0)\\
\Leftrightarrow &T(\beta_1,\beta_2) \leq N(R(u,v)).
\end{aligned}$$

Using the property given above, the following holds.

**Proposition 3.** *Let $T$ be an IMTL t-norm and let $I$ be its R-implicator. The granules from $\underline{apr}_R^{\min,I}(A)$ and $\overline{apr}_R^{\max,T}(coA)$ are disjoint (analogously, the granules from $\underline{apr}_R^{\min,I}(coA)$ and $\overline{apr}_R^{\max,T}(A)$ are disjoint too).*

*Proof.* Using Proposition 2, we have to prove that for all $u, v \in U$ it holds that

$$T(\underline{\mathrm{apr}}_R^{\min,I}(A)(u), \overline{\mathrm{apr}}_R^{\max,T}(coA)(v)) \leq N(R(u,v)).$$

Using the object monotonicity property [11] combined with the residuation principle we have the following

$$\begin{aligned}
&R(u,v) \leq I(\overline{\mathrm{apr}}_R^{\max,T}(coA)(v), \overline{\mathrm{apr}}_R^{\max,T}(coA)(u))\\
\Leftrightarrow &T(R(u,v), \overline{\mathrm{apr}}_R^{\max,T}(coA)(v)) \leq \overline{\mathrm{apr}}_R^{\max,T}(coA)(u)\\
\Leftrightarrow &\overline{\mathrm{apr}}_R^{\max,T}(coA)(v) \leq I(R(u,v), \overline{\mathrm{apr}}_R^{\max,T}(coA)(u))\\
\Leftrightarrow &\overline{\mathrm{apr}}_R^{\max,T}(coA)(v) \leq I(N(\overline{\mathrm{apr}}_R^{\max,T}(coA)(u)), N(R(u,v)))\\
\Leftrightarrow &\overline{\mathrm{apr}}_R^{\max,T}(coA)(v) \leq I(\underline{\mathrm{apr}}_R^{\min,I}(A)(u), N(R(u,v)))\\
\Leftrightarrow &T(\underline{\mathrm{apr}}_R^{\min,I}(A)(u), \overline{\mathrm{apr}}_R^{\max,T}(coA)(v)) \leq N(R(u,v)).
\end{aligned}$$

The third equivalence holds because of Eq. (1), the fourth equivalence holds because of the duality property of the fuzzy rough approximations [11], while the other equivalences are a consequence of the residuation property.

Consider now a binary classification problem, i.e. we distinguish two classes in $U$: $A$ and $coA$. Since $A$ and $coA$ are now both crisp sets, we can simplify the approximations (3) as:

$$\begin{aligned}
\underline{\mathrm{apr}}_R^{\min,I}(A)(u) &= \min(N(R(u,v)); v \in coA),\\
\overline{\mathrm{apr}}_R^{\max,T}(A)(u) &= \max(R(u,v); v \in A).
\end{aligned}$$

Let $w$ be an element for which the minimum of the first expression above is reached, i.e., $\underline{\mathrm{apr}}_R^{\min,I}(A)(u) = N(R(u,w))$. Since it holds that $\overline{\mathrm{apr}}_R^{\max,T}(coA)(w) = \max(R(w,v); v \in coA) = R(w,w) = 1$, we have that

$$T(\underline{\mathrm{apr}}_R^{\min,I}(A)(u), \overline{\mathrm{apr}}_R^{\max,T}(coA)(w)) = N(R(u,w)).$$

In this case, the granules are not only disjoint but we may say that they touch each other. From the perspective of binary classification, having $T$-disjoint granules guarantees that no overlapping rules will be generated during the rule induction procedure. Moreover, when additionally granules touch each other, the probability that the induced rules will cover the space in a proper way increased. Consider the extreme case where we have a set of granules for which all parameters $\beta$ are equal to 0. All these granules are pairwise $T$-disjoint, but they do not cover any space. By having two touching granules, we ensure that the space between them is maximally covered.

Figures 1 and 2 illustrate different positions of granules, in one and two dimensions respectively.

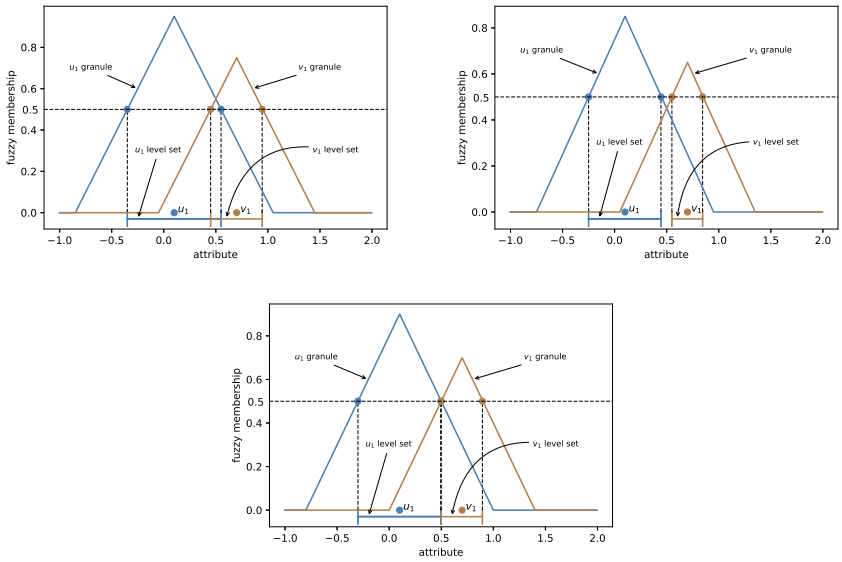


Fig. 1: Granules in one dimension

In Figure 1, objects are represented using one conditional attribute. There are two objects $u_1$ and $v_1$ that belong to different classes; their granules are fuzzy sets of triangular shape, and we also depict their 0.5-level sets. The granules are constructed using triangular similarity (2) and the Łukasiewicz $t$-norm. In the upper left figure, the granules overlap (i.e., they are not $T$-disjoint), in the upper right figure, they are $T$-disjoint, while in the lower figure they touch each other. It is easy to verify that in this case, the 0.5-level sets follow the relation between granules, i.e., if the granules overlap, then the level sets overlap, if the granules are $T$-disjoint, then the level sets are disjoint and if the granules touch each

other, then the level sets also touch each other (the type of touching depends on the dimension).

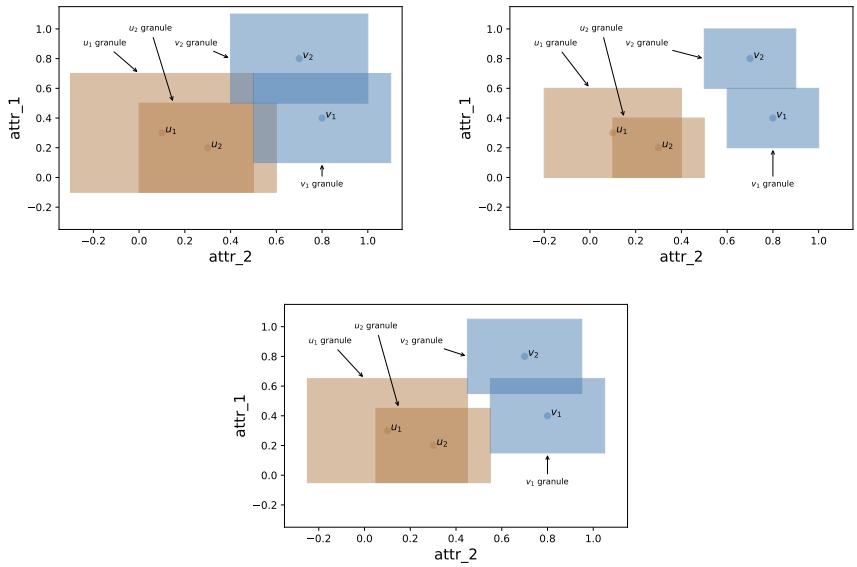


Fig. 2: Granules in two dimensions

The use of level sets is necessary to visualize the granules in the case of two dimensions. In Figure 2, we have four objects from two classes, described by two conditional attributes; $u_1$ and $u_2$ are from one class and $v_1$ and $v_2$ are from the other class. Their granules are represented in the figure using 0.5-level sets, which are of rectangular shape. Again, we can distinguish granules from different classes that overlap (upper left), are disjoint (upper right) and touch each other (lower). In this case, the levels sets of touching granules share an edge. We can notice that touching granules cover more space between objects than disjoint ones, which highlights the importance of the touching condition.

Bearing in mind the properties of $T$-disjointness and touching granules, we want to construct fuzzy sets other than the lower and upper fuzzy rough approximations that satisfy these properties. We first investigate the binary classification case. Denote the new fuzzy set to be constructed by $\Lambda = \{(u, \lambda_u); u \in U\}$. Let $u$ and $v$ be two objects from $U$. If they are from different classes, we want their corresponding granules to be $T$-disjoint, i.e.,

$$T(\lambda_u, \lambda_v) \leq N(R(u, v)).$$

Other than this, we do not impose any condition on the granules at this point. We now consider the equivalence relation $S$ on $U$ defined as $S(u, v) = 1$ if $u$

and $v$ are from the same decision class, and $S(u,v) = 0$ otherwise. If $u$ and $v$ are from a different decision class then $I(R(u,v), S(u,v)) = N(R(u,v))$ while $I(R(u,v), S(u,v)) = 1$ otherwise. Hence, the $T$-disjointness condition (5) for any two elements from $U$ may be reformulated as

$$T(\lambda_u, \lambda_v) \leq I(R(u,v), S(u,v)). \tag{6}$$

In the previous step, we have replaced the decision classes from the binary classification with relation $S$. Note that relation $S$ corresponds to the one mentioned in the Introduction; it is a replacement for the fuzzy set $A$ from the fuzzy rough approximations (3). It also explains the "relation-based" part in the name of RFGA.

The next step is to include the property of touching granules in the construction of $\Lambda$. Concretely, for any object $u \in U$, we want to have an object $w \in U$ whose granule is touched by that of $u$. It is not always possible to have that $T(\lambda_u, \lambda_w) = I(R(u,w), S(u,w))$. In general, if the granule of $w$ is touched by the granule of $u$, it means that for fixed $\lambda_w$, $\lambda_u$ should be the largest possible value for which (6) holds. More formally:

$$\begin{aligned} &\lambda_u = \sup\{\beta; T(\lambda_w, \beta) \leq I(R(u,w), S(u,w))\} \\ \Leftrightarrow &\lambda_u = I(\lambda_w, I(R(u,w), S(u,w))). \end{aligned}$$

If we apply the residuation property to (6) we have that

$$\forall v \in U, \quad \lambda_u \leq I(\lambda_v, I(R(u,v), S(u,v))).$$

This further implies

$$I(\lambda_w, I(R(u,w), S(u,w))) = \min_{v \in U} I(\lambda_v, I(R(u,v), S(u,v))),$$

which leads to the conclusion

$$\lambda_u = \min_{v \in U} I(\lambda_v, I(R(u,v), S(u,v))). \tag{7}$$

With (7), we have reached the system of equations that defines $\Lambda$. Feasible solutions of this system of equations yield granules that satisfy the properties of being $T$-disjoint and touching each other. This is the formal definition of the relation-based fuzzy granular approximation (RFGA).

### 3.1 Extension beyond binary classification

The definition of RFGA was given for the binary classification problem. The final expression contains relation $S$ that distinguishes between two classes. However, relation $S$ can be extended to a general equivalence relation in the case of multi-class classification problem. Moreover, it can be a fuzzy relation too. In the case of the regression task, a relation $S$ may be a measure of similarity between values of the decision attribute. In the case of $S$ being a fuzzy relation, the meaning of granules and the properties we mentioned above will be different. That meaning is an open question and it will be a part of our future research. From now on, we use the notation: $M(u,v) = I(R(u,v), S(u,v))$.

## 4 Calculation

We have developed the system of equations (7) that describes RFGA $\Lambda$. The question is, how to obtain a feasible solution of that system being useful in practice.

We have seen that (7) has a solution in the case of binary classification. That solution can be constructed using lower and upper fuzzy rough approximations. First, we generalize that result.

**Proposition 4.** *If $I$ is an $R$-implicator of some left-continuous $t$-norm, then (7) has a solution.*

*Proof.* We construct a feasible solution. Let $u_1, \ldots, u_n$ be an ordering of objects from $U$. We apply the following procedure.

1) $\lambda_{u_1}$ is a random value from $[0, 1]$.
2) For $1 < i \leq n$, $\lambda_{u_i} = \min_{j<i} I(\lambda_{u_j}, M(u_i, u_j))$.

The touching property is obvious from the construction. We have to prove the granularity property. Let $u_i$ and $u_k$ be two objects for which $k < i$. Since $\lambda_{u_i} = \min_{j<i} I(\lambda_{u_j}, M(u_i, u_j))$, it holds that $\lambda_{u_i} \leq I(\lambda_{u_k}, M(u_i, u_k))$. From the residuation property, this is equivalent to $T(\lambda_{u_i}, \lambda_{u_k}) \leq M(u_i, u_k)$.

However, the solution from the previous theorem heavily depends on the ordering of objects which may lead to quite imbalanced solutions. In the imbalanced solutions, some fuzzy values from $\Lambda$ have extremely high values (close to 1) while others have extremely low values (close to 0). A balanced solution has yet to be defined formally, but the main intuition may be seen on Figure 3. On the left image, we have an example of imbalanced solution where the fuzzy values from $\Lambda$ are calculated using lower approximation for $u_1$ and $u_2$ and upper approximation for $v_1$ and $v_2$. The imbalance comes from the fact that the space between the objects from different classes is covered mostly by the granules of $v_1$ and $v_2$. On the right image, we have an example of a more balanced solution where the space in-between the classes is covered roughly equally by the granules from different classes. The values of $\Lambda$ on the right image are calculated using the method explained below.

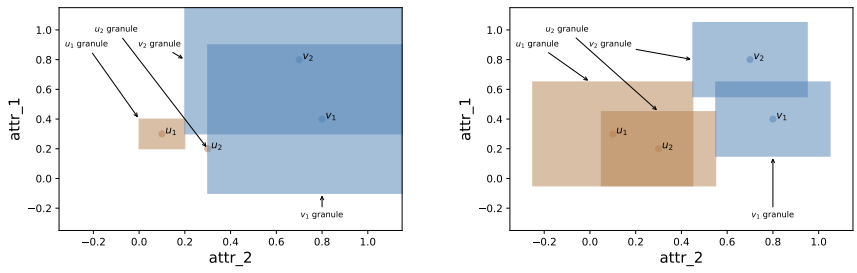


Fig. 3: Imbalanced vs. balanced granules

In general, finding a proper solution of (7) is an open question. One option is to find values that satisfy (6) and that are an optimal solution of some optimization problem. The optimality of such solution guarantees that (7) is satisfied. In such optimization problem, (6) will be the set of constraints while we have to choose a proper objective function. We develop such optimization problem for continuous Archimedean $t$-norms and their $R$-implicators. For these fuzzy connectives, (7) becomes

$$\begin{aligned}
&\lambda_u = \min_{v\in U} f^{-1}(\max(f(M(u,v)) - f(\lambda_v), 0)) \\
\Leftrightarrow &f(\lambda_u) = \max_{v\in U} \max(f(M(u,v)) - f(\lambda_v), 0) \\
\Leftrightarrow &f(\lambda_u) = \max(\max_{v\in U} f(M(u,v)) - f(\lambda_v), 0),
\end{aligned}$$

for some decreasing generator $f$. We introduce new variables $\forall u \in U$, $\alpha_u = f(\lambda_u)$ and $\forall u, v \in U$, $A(u,v) = f(M(u,v))$. Since $f$ is a positive function, $\alpha_u$ values are also positive. Because of that, the last equation becomes

$$\alpha_u = \max(\max_{v\in U}(A(u,v) - \alpha_v), 0). \tag{8}$$

So, in the case of continuous Archimedean $t$-norms, we have to solve (8). The set of constraints for our optimization problem that reflects (6) is

$$\alpha_u \geq A(u,v) - \alpha_v \Leftrightarrow \alpha_u + \alpha_v \geq A(u,v).$$

The next step is to select the appropriate objective function, the optimization of which would lead to a solution of (8). To achieve that, for a fixed $u \in U$, at least one inequality $\alpha_u + \alpha_v \geq A(u,v)$ for $v \in U$ has to be an equality. This can be accomplished by minimizing the values of $\alpha_v$, i.e. minimizing their sum $\sum_{u\in U} \alpha_u$. We now get the following optimization problem:

$$\begin{aligned}
&\text{minimize} && \sum_{u\in U} \alpha_u \\
&\text{subject to} && \alpha_u + \alpha_v \geq A(u,v), && u, v \in U \\
& && \alpha_u \geq 0, && u \in U.
\end{aligned}$$

This optimization problem can be solved efficiently using linear programming techniques. The efficiency was also a reason why we chose a linear objective function. While this way of calculation may produce imbalanced solutions, we are able to avoid it by introducing a constraint that the largest value $\alpha_u$ is as smallest as possible. This is done by adding one additional variable $a$ and reformulating the optimization problem as follows:

$$\begin{aligned}
&\text{maximize} && \sum_{u\in U} \alpha_u + Ca \\
&\text{subject to} && \alpha_u + \alpha_v \geq A(u,v), && u, v \in U, \\
& && \alpha_u \geq 0, && u \in U, \\
& && a \geq \alpha_u, && u \in U,
\end{aligned} \tag{9}$$

where $C$ is a significantly large constant. Obviously, $a = \min_{u \in U} \alpha_u$ in the optimal solution. If it would hold that $a > \min_{u \in U} \alpha_u$, then taking $a - \epsilon$ for some $\epsilon > 0$ would give us a smaller value of the objective, contradicting the optimality. With the new constraint, we avoid imbalanced solutions by eliminating extremely large values of $\alpha_u$, which further leads to the elimination of the extremely small values of $\alpha_u$ due to the other constraints. We still have to prove that the previous linear program indeed solves (7). Since the $T$-disjointness property is satisfied by the constraints, we have to prove the touching property. We have the following proposition.

**Proposition 5.** *For every $\alpha_u$, $u \in U$ that is a solution of (9) it holds that $\alpha_u = 0$ or $\exists v \in U$, $\alpha_u + \alpha_v = A(u, v)$.*

*Proof.* Let $u \in U$ be an object for which the proposition is not satisfied. Then for every $v \in U$ it holds that $\alpha_u + \alpha_v > A(u, v)$ and $\alpha_u > 0$. If we take $\epsilon = \max(\max_{v \in U}(A(u, v) - \alpha_v), 0)$ then replacing $\alpha_u$ with $\alpha_u - \epsilon$ decreases the value of the objective. That contradicts the assumption of optimality.

If $\alpha_u + \alpha_v = A(u, v)$, then $\lambda_u = I(\lambda_v, M(u, v))$. If $\alpha_u = 0$, then $\lambda_u = 1$ and $u$ is touching every other granule, i.e., $\lambda_u = I(\lambda_v, M(u, v))$ for all $v \in U$.

As we may notice, the derivation and proofs of correctness do not depend on the properties of relation $S$ therefore, we may do the calculation for any (crisp or fuzzy) relation $S$.

## 5 Application of RFGA in Machine Learning and for making predictions

In the previous sections, we have defined the RFGA and proposed a way to calculate it. The following step is to investigate its use in Data Analysis and Machine Learning. We mentioned rule induction before, but for now we will hold off on presenting rule induction algorithms and will come back to it in our future work. The other possible use of the RFGA that we will briefly discuss is the Local Model Adjustment.

### 5.1 Model mimicking

The RFGA alone is not suitable for the general prediction tasks. The RFGA does not change the labels on the training set. It calculates set $\Lambda$ respecting the initial decision values but it does not change them. Otherwise, relation $S$ would not be a constant but a variable. Therefore, it is desirable to combine the RFGA with some Machine Learning model. The idea here is to apply that ML model on raw data, relabel the data (change the decision values) based on the predictions of the model and to apply the RFGA on the relabeled data. After that, the RFGA should replace the original Machine Learning model as a predictive model. Making predictions with RFGA is done in the following way. Assume that we have an unseen object $u$ which decision value we want to

predict. Let $d(u)$ denote a possible decision value of object $u$ and let $d_{pred}(u)$ be the actual prediction. Let $D$ be a domain of the decision attribute (if we have a classification then $D$ is a finite set, while if we deal with a regression then $D$ is a set of real numbers). The prediction will be the value from domain $D$ for which the RFGA value on $u$ is maximal, i.e.

$$d_{pred}(u) = \arg \max_{d(u) \in D} \min_{v \in U_{train}} I(\lambda_v, I(R(u,v), S(u,v))),$$

where $U_{train}$ is now the set of training objects. The only value that depends on $d(u)$ in the expression is $S(u,v)$ since it measures the similarity between $u$ and $v$ on the decision attribute i.e. the one that we have to predict. If $D$ is finite, making predictions is easy. For each value in $D$, we calculate the value of the expression (the RFGA value), and the prediction will be the value from D for which the value of the expression the largest possible one. If $D$ is the set of real numbers, the same methodology is infeasible. In that case, the mapping

$$d(u) \to \min_{v \in U_{train}} I(\lambda_v, I(R(u,v), S(u,v))),$$

is a mapping from real numbers to real numbers. Looking for the maximum of such mapping has to include some iterative methods like the gradient descent. The combination of those iterative methods with the RFGA is still an open question.

In short, we replaced an original ML model with the RFGA that is able to make predictions and with that RFGA we mimic the original ML model. One possible application of the model mimicking is the Local Model Adjustment (LMA)

### 5.2 Local Model Adjustment

After a model is mimicked with the RFGA, we have a model that assigns a fuzzy value to each training object ($\lambda_u$ for $u \in U$). This fuzzy value, from the perspective of making predictions, may be seen as a measure of an influence of underlying object in making predictions. If the fuzzy value is larger, then the influence is larger. That may be seen in the following way. Implicator $I$ is a non-increasing function in its first argument, which implies that if $\lambda_v$ is increased then $I(\lambda_v, M(u,v))$ is decreased, which makes it closer to the minimum i.e., it influences more the prediction. Bearing this in mind, RFGA gives us the ability to inspect the model locally i.e. to examine the influence of a particular training object in making predictions. For example, we have an RFGA prediction model which works fine in majority of cases i.e. the predictions made on unseen objects coincides with reality (e.g. a model trained to accept or reject loan proposals indeed accepts those with a good credit rating and rejects those with a bad one) except when the condition attributes take some small set of values where the performance is poor. Adjusting the RFGA fuzzy values on objects that are close to that specific small set of values, we are maybe able to improve the performance locally while not affecting the remaining parts of the model. This can be also seen as the model debugging.

## 6 Some experiments

In this section, we want to show how good is the RFGA in model mimicking. Since we are still in the development of methods for regression, the following results are related to the classification problems. For the following experiments, Łukasiewicz $t$-norm is used with the corresponding R-implicator. For the fuzzy relation $R$, the triangular similarity (2) is used.

In Table 2 we provide the results of the model mimicking approach using the approach provided in Sections 5 where the RFGA is calculated using the linear programming method from Section 4. We have collected datasets, which details are in the table, and applied model mimicking as described in Subsection 4.1. The 5-fold cross-validation is applied. The first three columns are the details of the datasets, the fourth one is the ML model applied on data prior to the RFGA. The following two are the average balanced accuracies [1] of the ML model on the train and test data, while the last two columns are the average balanced accuracies of the RFGA model on the train and test data. In the column "model" the abbreviations have the following meaning. RF($a$, $b$) stands for the Random Forest [8] classifier with $a$ decision trees where the maximal depth of every decision tree is $b$. SVM($c$) stands for the Support Vector Machine [2] with the Gaussian kernel with parameter $c$. Prior to the model application on data, a feature selection procedure based on the Random Forests was applied to reduce the dimensionality.

| name | n_col | n_att | n_class | model | train | test | RFGA_train | RFGA_test |
|---|---|---|---|---|---|---|---|---|
| ionosphere | 351 | 32 | 2 | RF(10, 4) | .931 | .905 | .931 | .86 |
| mammographic | 830 | 5 | 2 | RF(15, 4) | .833 | .824 | .833 | .822 |
| monk-2 | 432 | 6 | 2 | RF(15, 4) | .973 | .974 | .973 | .973 |
| wisconsin | 683 | 9 | 2 | RF(15, 4) | .970 | .962 | .970 | .964 |
| wdbc | 569 | 30 | 2 | RF(15,4) | .975 | .941 | .975 | .938 |
| dermatology | 358 | 34 | 6 | RF(15,4) | .958 | .947 | .958 | .927 |
| wowel | 990 | 11 | 11 | SVM(10) | .998 | .983 | .998 | .925 |
| iris | 150 | 4 | 3 | SVM(0.5) | .96 | .954 | .96 | .952 |

Table 2: RFGA performance in model mimicking

The experiments are implemented in programming language PYTHON with packages NUMPY [7] for the numeric calculations and SCIKIT-LEARN [14] for Machine Learning. The seed for the random number generator in NUMPY was set to 123.

We can see that for the majority of datasets, the RFGA gives a similar performance on the test sets as the underlying ML model. There are some exceptions like the "ionosphere" and "wowel" datasets where the drop in performance on test sets is around 6%. While such drop is not significant, we are still eager to improve it. We have noticed that the feature selection prior to the modeling was

necessary since the drop in performance would be larger without it. We may conclude that the RFGA may be sensitive if there are many conditional attributes which do not contribute to the prediction.

## 7 Conclusion

We have introduced a formal definition of the relation-based fuzzy granular approximation (RFGA) in order to generalize the granularity properties of the fuzzy rough sets in the case when fuzzy relations are used on the decision attribute. After we provided the formal definition, we explain how to calculate it using linear programming methods. Later on, we discussed possible applications in the Machine Learning problems and we provided some experimental results to illustrate how good is the RFGA in mimicking a Machine Learning model. We list some possibilities for the future research.

- We are still missing a proper interpretation of fuzzy set $\Lambda$ from the fuzzy logic perspective. Despite having nice properties regarding granularity and adjustability, we want to understand better the real meaning of fuzzy set $\Lambda$.
- How to properly apply the approach on the regression problems? While the calculation of $\Lambda$ is not an issue, the real problem is how to make predictions.
- We had mentioned imbalanced solutions but we did not formally define it. One of the next step will be to derive a definition of a balanced solution such that we can measure it and control it in the better way.
- RFGA system of equations has many solutions while the linear program we proposed returns only one. We want to be able to obtain more solutions and to combine them to get better performance.
- We were briefly discussing about the Local Model Adjustment but we still have to apply it in practice. The question here is how set up a benchmark in order to apply the RFGA for the LMA. We have to collect data which we understand well and on which we are able to detect that some fitted model performs well in general but performs poorly on a small isolated region.
- The whole motivation for the granularity arises from its potential application in rule induction. Since now we have defined RFGA, we are able to construct and test rule induction algorithms on different datasets. This will be the key part of our future research.

## Acknowledgements

This work was supported by the Odysseus program of the Research Foundation-Flanders. The research of Roman Słowiński was supported by the Polish Ministry of Education and Science, grant no. 0311/SBAD/0709.

## References

1. Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M.: The balanced accuracy and its posterior distribution. In: 2010 20th international conference on pattern recognition. pp. 3121–3124. IEEE (2010)

2. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
3. Degang, C., Yongping, Y., Hui, W.: Granular computing based on fuzzy similarity relations. Soft Computing **15**(6), 1161–1172 (2011)
4. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. International Journal of General System **17**(2-3), 191–209 (1990)
5. Greco, S., Matarazzo, B., Slowinski, R.: Fuzzy extension of the rough set approach to multicriteria and multiattribute sorting. In: Preferences and decisions under incomplete knowledge, pp. 131–151. Springer (2000)
6. Grzymala-Busse, J.W.: Lers-a system for learning from examples based on rough sets. In: Intelligent decision support, pp. 3–18. Springer (1992)
7. Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., et al.: Array programming with numpy. Nature **585**(7825), 357–362 (2020)
8. Ho, T.K.: Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition. vol. 1, pp. 278–282. IEEE (1995)
9. Jensen, R., Cornelis, C., Shen, Q.: Hybrid fuzzy-rough rule induction and feature selection. In: 2009 IEEE International Conference on Fuzzy Systems. pp. 1151–1156. IEEE (2009)
10. Klement, E.P., Mesiar, R., Pap, E.: Triangular norms, vol. 8. Springer Science & Business Media (2013)
11. Palangetić, M., Cornelis, C., Greco, S., Słowiński, R.: Fuzzy extensions of the dominance-based rough set approach. International Journal of Approximate Reasoning **129**, 1–19
12. Palangetić, M., Cornelis, C., Greco, S., Słowiński, R.: Granular representation of owa-based fuzzy rough sets. Submitted to Fuzzy Sets and Systems (2021)
13. Pawlak, Z.: Rough sets. International journal of computer & information sciences **11**(5), 341–356 (1982)
14. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)
15. Stefanowski, J.: On combined classifiers, rule induction and rough sets. In: Transactions on rough sets VI, pp. 329–350. Springer (2007)
16. Yao, Y.: Rough sets, neighborhood systems and granular computing. In: Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No. 99TH8411). vol. 3, pp. 1553–1558. IEEE (1999)
17. Zadeh, L.: Fuzzy sets. Information and Control **8(3)**, 338–353 (1965)
18. Zhao, S., Tsang, E.C., Chen, D., Wang, X.: Building a rule-based classifier—a fuzzy-rough set approach. IEEE Transactions on Knowledge and Data Engineering **22**(5), 624–638 (2009)