Fuzzy-Rough Nearest Neighbour Classification

Richard Jensen[1] and Chris Cornelis[2]

[1] Dept. of Comp. Sci., Aberystwyth University, Ceredigion, SY23 3DB, Wales, UK
rkj@aber.ac.uk
[2] Dept. of Appl. Math. and Comp. Sci., Ghent University, Gent, Belgium
Chris.Cornelis@UGent.be

Abstract. A new fuzzy-rough nearest neighbour (FRNN) classification algorithm is presented in this paper, as an alternative to Sarkar's fuzzy-rough ownership function (FRNN-O) approach. By contrast to the latter, our method uses the nearest neighbours to construct lower and upper approximations of decision classes, and classifies test instances based on their membership to these approximations. In the experimental analysis, we evaluate our approach with both classical fuzzy-rough approximations (based on an implicator and a t-norm), as well as with the recently introduced vaguely quantified rough sets. Preliminary results are very good, and in general FRNN outperforms FRNN-O, as well as the traditional fuzzy nearest neighbour (FNN) algorithm.

Keywords: Fuzzy-rough sets, nearest neighbour algorithms, classification.

1 Introduction

Lately there has been great interest in developing methodologies which are capable of dealing with imprecision and uncertainty, and the resounding amount of research currently being done in the areas related to fuzzy [30] and rough sets [18] is representative of this. The success of rough set theory is due in part to three aspects of the theory. Firstly, only the facts hidden in data are analysed. Secondly, no additional information about the data is required for data analysis such as thresholds or expert knowledge on a particular domain. Thirdly, it finds a minimal knowledge representation for data. As rough set theory handles only one type of imperfection found in data, it is complementary to other concepts for the purpose, such as fuzzy set theory. The two fields may be considered analogous in the sense that both can tolerate inconsistency and uncertainty - the difference being the type of uncertainty and their approach to it; fuzzy sets are concerned with vagueness, rough sets are concerned with indiscernibility.

Many relationships have been established and more so, most of the recent studies have concluded at this complementary nature of the two methodologies, especially in the context of granular computing. Therefore, it is desirable to extend and hybridize the underlying concepts to deal with additional aspects of data imperfection. Such developments offer a high degree of flexibility and provide robust solutions and advanced tools for data analysis.

J.F. Peters et al. (Eds.): Transactions on Rough Sets XIII, LNCS 6499, pp. 56–72, 2011.

The K-nearest neighbour (KNN) algorithm [9] is a well-known classification technique that assigns a test object to the decision class most common among its K nearest neighbours, i.e., the K training objects that are closest to the test object. An extension of the KNN algorithm to fuzzy set theory (FNN) was introduced in [17]. It allows partial membership of an object to different classes, and also takes into account the relative importance (closeness) of each neighbour w.r.t. the test instance. However, as Sarkar correctly argued in [22], the FNN algorithm has problems dealing adequately with insufficient knowledge. In particular, when every training pattern is far removed from the test object, and hence there are no suitable neighbours, the algorithm is still forced to make clear-cut predictions. This is because the predicted membership degrees to the various decision classes always need to sum up to 1.

To address this problem, Sarkar [22] introduced a so-called fuzzy-rough ownership function that, when plugged into the conventional FNN algorithm, produces class confidence values that do not necessarily sum up to 1. However, this method (called FRNN-O throughout this paper) does not refer to the main ingredients of rough set theory, i.e., lower and upper approximation. In this paper, therefore, we present an alternative approach, which uses a test object's nearest neighbours to construct the lower and upper approximation of each decision class, and then computes the membership of the test object to these approximations. The method is very flexible, as there are many options to define the fuzzy-rough approximations, including the traditional implicator/t-norm based model [21], as well as the vaguely quantified rough set (VQRS) model [6], which is more robust in the presence of noisy data.

This paper is structured as follows. Section 2 provides necessary details for fuzzy rough set theory, while Section 3 is concerned with the existing fuzzy (-rough) NN approaches. Section 4 outlines our algorithm, while comparative experimentation on a series of classification and prediction problems is provided in Section 5. The paper is concluded in section 6.

2 Hybridization of Rough Sets and Fuzzy Sets

2.1 Rough Set Theory

Rough set theory (RST) [18] provides a tool by which knowledge may be extracted from a domain in a concise way; it is able to retain the information content whilst reducing the amount of knowledge involved. Central to RST is the concept of indiscernibility. Let $(\mathbb{U}, \mathbb{A})$ be an information system, where $\mathbb{U}$ is a non-empty set of finite objects (the universe of discourse) and $\mathbb{A}$ is a non-empty finite set of attributes such that $a : \mathbb{U} \rightarrow V_a$ for every $a \in \mathbb{A}$. V_a is the set of values that attribute a may take. With any $B \subseteq \mathbb{A}$ there is an associated equivalence relation R_B:

$$R_B = \{(x, y) \in \mathbb{U}^2 | \forall a \in B,\ a(x) = a(y)\} \tag{1}$$

If $(x, y) \in R_B$, then x and y are indiscernible by attributes from B. The equivalence classes of the B-indiscernibility relation are denoted $[x]_B$. Let $A \subseteq \mathbb{U}$. A

can be approximated using the information contained within B by constructing the B-*lower* and B-*upper* approximations of A:

$$R_B{\downarrow}A = \{x \in \mathbb{U} \mid [x]_B \subseteq A\} \tag{2}$$

$$R_B{\uparrow}A = \{x \in \mathbb{U} \mid [x]_B \cap A \neq \emptyset\} \tag{3}$$

The tuple $\langle R_B{\downarrow}A, R_B{\uparrow}A\rangle$ is called a rough set.

A *decision system* $(X, \mathcal{A} \cup \{d\})$ is a special kind of information system, used in the context of classification, in which d ($d \notin \mathcal{A}$) is a designated attribute called the decision attribute. Its equivalence classes $[x]_{R_d}$ are called decision classes. The set of decision classes is denoted $\mathcal{C}$ in this paper. Given $B \subseteq \mathcal{A}$, the B-positive region POS_B contains those objects from X for which the values of B allow to predict the decision class unequivocally:

$$POS_B = \bigcup_{x \in X} R_B{\downarrow}[x]_{R_d} \tag{4}$$

Indeed, if $x \in POS_B$, it means that whenever an object has the same values as x for the attributes in B, it will also belong to the same decision class as x. The predictive ability w.r.t. d of the attributes in B is then measured by the following value (degree of dependency of d on B):

$$\gamma_B = \frac{|POS_B|}{|X|} \tag{5}$$

$(X, \mathcal{A} \cup \{d\})$ is called *consistent* if $\gamma_{\mathcal{A}} = 1$. A subset B of $\mathcal{A}$ is called a *decision reduct* if it satisfies $POS_B = POS_{\mathcal{A}}$, i.e., B preserves the decision making power of $\mathcal{A}$, and moreover it cannot be further reduced, i.e., there exists no proper subset B' of B such that $POS_{B'} = POS_{\mathcal{A}}$. If the latter constraint is lifted, i.e., B is not necessarily minimal, we call B a decision superreduct.

2.2 Fuzzy Set Theory

Fuzzy set theory [30] allows that objects belong to a set, or couples of objects belong to a relation, to a given degree. Recall that a fuzzy set in X is an $X \to [0,1]$ mapping, while a fuzzy relation in X is a fuzzy set in $X \times X$. For all y in X, the R-foreset of y is the fuzzy set Ry defined by

$$Ry(x) = R(x, y) \tag{6}$$

for all x in X. If R is a reflexive and symmetric fuzzy relation, that is,

$$R(x, x) = 1 \tag{7}$$

$$R(x, y) = R(y, x) \tag{8}$$

hold for all x and y in X, then R is called a fuzzy tolerance relation. For a fuzzy tolerance relation R, we call Ry the fuzzy tolerance class of y.

For fuzzy sets A and B in X, $A \subseteq B \iff (\forall x \in X)(A(x) \leq B(x))$. If X is finite, the cardinality of A is calculated by

$$|A| = \sum_{x \in X} A(x). \tag{9}$$

Fuzzy logic connectives play an important role in the development of fuzzy rough set theory. We therefore recall some important definitions. A triangular norm (t-norm for short) $\mathcal{T}$ is any increasing, commutative and associative $[0,1]^2 \rightarrow [0,1]$ mapping satisfying $\mathcal{T}(1,x) = x$, for all x in $[0,1]$. In this paper, we use $\mathcal{T}_M$ and $\mathcal{T}_L$ defined by $\mathcal{T}_M(x,y) = \min(x,y)$ and $\mathcal{T}_L(x,y) = \max(0, x+y-1)$ (Łukasiewicz t-norm), for x, y in $[0,1]$. On the other hand, an implicator is any $[0,1]^2 \rightarrow [0,1]$-mapping $\mathcal{I}$ satisfying $\mathcal{I}(0,0) = 1, \mathcal{I}(1,x) = x$, for all x in $[0,1]$. Moreover we require $\mathcal{I}$ to be decreasing in its first, and increasing in its second component. The implicators used in this paper are $\mathcal{I}_M$ and $\mathcal{I}_L$ defined by $\mathcal{I}_M(x,y) = \max(1-x, y)$ (Kleene-Dienes implicator) and $\mathcal{I}_L(x,y) = \min(1, 1-x+y)$ (Łukasiewicz implicator) for x, y in $[0,1]$.

2.3 Fuzzy-Rough Set Theory

The process described above can only operate effectively with datasets containing discrete values. As most datasets contain real-valued attributes, it is necessary to perform a discretization step beforehand. A more intuitive and flexible approach, however, is to model the approximate equality between objects with continuous attribute values by means of a fuzzy relation R in $\mathbb{U}$, i.e., a $\mathbb{U} \rightarrow [0,1]$ mapping that assigns to each couple of objects their degree of similarity. In general, it is assumed that R is at least a fuzzy tolerance relation, that is, $R(x,x) = 1$ and $R(x,y) = R(y,x)$ for x and y in $\mathbb{U}$. Given y in $\mathbb{U}$, its foreset Ry is defined by $Ry(x) = R(x,y)$ for every x in $\mathbb{U}$.

Given a fuzzy tolerance relation R and a fuzzy set A in $\mathbb{U}$, the lower and upper approximation of A by R can be constructed in several ways. A general definition [7,21] is the following:

$$(R{\downarrow}A)(x) = \inf_{y \in \mathbb{U}} \mathcal{I}(R(x,y), A(y)) \tag{10}$$

$$(R{\uparrow}A)(x) = \sup_{y \in \mathbb{U}} \mathcal{T}(R(x,y), A(y)) \tag{11}$$

Here, I is an implicator and T a t-norm. When A is a crisp (classical) set and R is an equivalence relation in $\mathbb{U}$, the traditional lower and upper approximation are recovered.

Just like their crisp counterparts, formulas (10) and (11) (henceforth called the FRS approximations) are quite sensitive to noisy values. That is, a change in a single object can result in drastic changes to the approximations (due to the use of sup and inf, which generalize the existential and universal quantifier, respectively). In the context of classification tasks, this behaviour may affect accuracy adversely. Therefore, in [6], the concept of vaguely quantified rough sets (VQRS) was introduced. It uses the linguistic quantifiers “most” and “some”, as opposed to the traditionally used crisp quantifiers "all" and "at least one", to decide to what extent an object belongs to the lower and upper approximation. Given a couple (Q_u, Q_l) of fuzzy quantifiers[1] that model "most" and "some", the lower and upper approximation of A by R are defined by

$$(R{\downarrow}^{Q_u} A)(y) = Q_u\left(\frac{|Ry \cap A|}{|Ry|}\right) = Q_u\left(\frac{\sum\limits_{x \in X} \min(R(x,y), A(x))}{\sum\limits_{x \in X} R(x,y)}\right) \tag{12}$$

$$(R{\uparrow}^{Q_l} A)(y) = Q_l\left(\frac{|Ry \cap A|}{|Ry|}\right) = Q_l\left(\frac{\sum\limits_{x \in X} \min(R(x,y), A(x))}{\sum\limits_{x \in X} R(x,y)}\right) \tag{13}$$

where the fuzzy set intersection is defined by the min t-norm and the fuzzy set cardinality by the sigma-count operation. As an important difference to (10) and (11), the VQRS approximations do not extend the classical rough set approximations, in a sense that when A and R are crisp, the lower and upper approximations may still be fuzzy.

2.4 Fuzzy-Rough Classification

Due to its recency, there have been very few attempts at developing fuzzy-rough set theory for the purpose of classification. Previous work has focused on using crisp rough set theory to generate fuzzy rulesets [14,23] but mainly ignores the direct use of fuzzy-rough concepts.

The induction of gradual decision rules, based on fuzzy-rough hybridization, is given in [12]. For this approach, new definitions of fuzzy lower and upper approximations are constructed that avoid the use of fuzzy logical connectives altogether. Decision rules are induced from lower and upper approximations defined for positive and negative relationships between credibility of premises and conclusions. Only the ordinal properties of fuzzy membership degrees are used. More recently, a fuzzy-rough approach to fuzzy rule induction was presented in [27], where fuzzy reducts are employed to generate rules from data. This method also employs a fuzzy-rough feature selection preprocessing step.

Also of interest is the use of fuzzy-rough concepts in building fuzzy decision trees. Initial research is presented in [2] where a method for fuzzy decision tree construction is given that employs the fuzzy-rough ownership function. This is used to define both an index of fuzzy-roughness and a measure of fuzzy-rough entropy as a node splitting criterion. Traditionally, fuzzy entropy (or its extension) has been used for this purpose. In [16], a fuzzy decision tree algorithm is proposed, based on fuzzy ID3, that incorporates the fuzzy-rough dependency function as a splitting criterion. A fuzzy-rough rule induction method is proposed in [13] for generating certain and possible rulesets from hierarchical data.

[1] By a fuzzy quantifier, we mean an increasing $[0,1] \to [0,1]$ mapping such that $Q(0) = 0$ and $Q(1) = 1$.

3 Fuzzy Nearest Neighbour Classification

The fuzzy K-nearest neighbour (FNN) algorithm [17] was introduced to classify test objects based on their similarity to a given number K of neighbours (among the training objects), and these neighbours' membership degrees to (crisp or fuzzy) class labels. For the purposes of FNN, the extent $C'(y)$ to which an unclassified object y belongs to a class C is computed as:

$$C'(y) = \sum_{x \in N} R(x,y)C(x) \tag{14}$$

where N is the set of object y's K nearest neighbours, obtained by calculating the fuzzy similarity between y and all training objects, and choosing the K objects that have highest similarity degree. $R(x,y)$ is the [0,1]-valued similarity of x and y. In the traditional approach, this is defined in the following way:

$$R(x,y) = \frac{||y-x||^{-2/(m-1)}}{\sum\limits_{j \in N} ||y-j||^{-2/(m-1)}} \tag{15}$$

where $|| \cdot ||$ denotes the Euclidean norm, and m is a parameter that controls the overall weighting of the similarity. Assuming crisp classes, Fig. 1 shows an application of the FNN algorithm that classifies a test object y to the class with the highest resulting membership. The idea behind this algorithm is that the degree of closeness of neighbours should influence the impact that their class membership has on deriving the class membership for the test object. The complexity of this algorithm for the classification of one test pattern is $O(|\mathbb{U}| + K \cdot |C|)$.

FNN($\mathbb{U}$,$\mathcal{C}$,y,K).
$\mathbb{U}$, the training data; $\mathcal{C}$, the set of decision classes;
y, the object to be classified; K, the number of nearest neighbours.

```
$N \leftarrow$ getNearestNeighbours($y$,$K$);
$\forall C \in \mathcal{C}$
    $C'(y) = \sum_{x \in N} R(x,y)C(x)$
output $\arg\max_{C \in \mathcal{C}} (C'(y))$
```

Fig. 1. The fuzzy KNN algorithm

Initial attempts to combine the FNN algorithm with concepts from fuzzy rough set theory were presented in [22,26]. In these papers, a fuzzy-rough ownership function is constructed that attempts to handle both "fuzzy uncertainty" (caused by overlapping classes) and "rough uncertainty" (caused by insufficient knowledge, i.e., attributes, about the objects). The fuzzy-rough ownership function τ_C of class C was defined as, for an object y,

$$\tau_C(y) = \frac{\sum\limits_{x \in \mathbb{U}} R(x,y)C(x)}{|\mathbb{U}|} \tag{16}$$

In this, the fuzzy relation R is determined by:

$$R(x,y) = \exp\left(-\sum_{a\in\mathbb{C}} \kappa_a (a(y)-a(x))^{2/(m-1)}\right) \tag{17}$$

where m controls the weighting of the similarity (as in FNN) and κ_a is a parameter that decides the bandwidth of the membership, defined as

$$\kappa_a = \frac{|\mathbb{U}|}{2\sum\limits_{x\in\mathbb{U}} ||a(y)-a(x)||^{2/(m-1)}} \tag{18}$$

$\tau_C(y)$ is interpreted as the confidence with which y can be classified to class C. The corresponding crisp classification algorithm, called FRNN-O in this paper, can be seen in Fig. 2. Initially, the parameter κ_a is calculated for each attribute and all memberships of decision classes for test object y are set to 0. Next, the weighted distance of y from all objects in the universe is computed and used to update the class memberships of y via equation (16). Finally, when all training objects have been considered, the algorithm outputs the class with highest membership. The algorithm's complexity is $O(|\mathbb{C}||\mathbb{U}| + |\mathbb{U}| \cdot (|\mathbb{C}| + |\mathcal{C}|))$.

By contrast to the FNN algorithm, the fuzzy-rough ownership function considers all training objects rather than a limited set of neighbours, and hence no decision is required as to the number of neighbours to consider. The reasoning behind this is that very distant training objects will not influence the outcome (as opposed to the case of FNN). For comparison purposes, the K-nearest neighbours version of this algorithm is obtained by replacing line (3) with $N \leftarrow$ getNearestNeighbours(y,K).

It should be noted that the algorithm does not use fuzzy lower or upper approximations to determine class membership. A very preliminary attempt to do so was described in [3]. However, the authors did not state how to use the upper and lower approximations to derive classifications.

FRNN-O($\mathbb{U}$,$\mathbb{C}$,$\mathcal{C}$,y).
$\mathbb{U}$, the training data; $\mathbb{C}$, the set of conditional features;
$\mathcal{C}$, the set of decision classes; y, the object to be classified.

```
(1)  $\forall a \in \mathbb{C}$
(2)      $\kappa_a = |\mathbb{U}|/2\sum_{x\in\mathbb{U}} ||a(y)-a(x)||^{2/(m-1)}$
(3)  $N \leftarrow |\mathbb{U}|$
(4)  $\forall C \in \mathcal{C}$, $\tau_C(y) = 0$
(5)  $\forall x \in N$
(6)      $d = \sum_{a\in\mathbb{C}} \kappa_a (a(y)-a(x))^2$
(7)      $\forall C \in \mathcal{C}$
(8)          $\tau_C(y) += \frac{C(x)\cdot\exp(-d^{1/(m-1)})}{|N|}$
(9)  output $\arg\max_{C\in\mathcal{C}} \tau_C(y)$
```

Fig. 2. The fuzzy-rough ownership nearest neighbour algorithm

4 Fuzzy-Rough Nearest Neighbour (FRNN) Algorithm

Figure 3 outlines our proposed algorithm, combining fuzzy-rough approximations with the ideas of the classical FNN approach. In what follows, FRNN-FRS and FRNN-VQRS denote instances of the algorithm where traditional, and VQRS, approximations are used, respectively. The rationale behind the algorithm is that the lower and the upper approximation of a decision class, calculated by means of the nearest neighbours of a test object y, provide good clues to predict the membership of the test object to that class.

In particular, if $(R{\downarrow}C)(y)$ is high, it reflects that all (most) of y's neighbours belong to C, while a high value of $(R{\uparrow}C)(y)$ means that at least one (some) neighbour(s) belong(s) to that class, depending on whether the FRS or VQRS approximations are used. A classification will always be determined for y due to the initialisation of τ to zero in line (2). To perform crisp classification, the algorithm outputs the decision class with the resulting best combined fuzzy lower and upper approximation memberships, seen in line (4) of the algorithm. This is only one way of utilising the information in the fuzzy lower and upper approximations to determine class membership, other ways are possible but are not investigated in this paper. The complexity of the algorithm is $O(|\mathcal{C}| \cdot (2|\mathbb{U}|))$.

FRNN($\mathbb{U}$,$\mathcal{C}$,y).
$\mathbb{U}$, the training data; $\mathcal{C}$, the set of decision classes;
y, the object to be classified.

```
(1)   $N \leftarrow$ getNearestNeighbors($y$,$K$)
(2)   $\tau \leftarrow 0$, $Class \leftarrow \emptyset$
(3)   $\forall C \in \mathcal{C}$
(4)      if $(((R{\downarrow}C)(y) + (R{\uparrow}C)(y))/2 \geq \tau)$
(5)         $Class \leftarrow C$
(6)         $\tau \leftarrow ((R{\downarrow}C)(y) + (R{\uparrow}C)(y))/2$
(7)   output $Class$
```

Fig. 3. The fuzzy-rough nearest neighbour algorithm - classification

Furthermore, the algorithm is dependent on the choice of the fuzzy tolerance relation R A general way of constructing R is as follows: given the set of conditional attributes $\mathbb{C}$, R is defined by

$$R(x,y) = \min_{a \in \mathbb{C}} R_a(x,y) \tag{19}$$

in which $R_a(x,y)$ is the degree to which objects x and y are similar for attribute a. Possible options include

$$R_a^1(x,y) = \exp\left(-\frac{(a(x)-a(y))^2}{2{\sigma_a}^2}\right) \tag{20}$$

$$R_a^2(x,y) = 1 - \frac{|a(x)-a(y)|}{|a_{\max} - a_{\min}|} \tag{21}$$

```
FRNN2($\mathbb{U}$,$d$,$y$).
$\mathbb{U}$, the training data; $d$, the decision feature;
$y$, the object to be classified.

$N \leftarrow$ getNearestNeighbors($y$,$K$)
$\tau_1 \leftarrow 0$, $\tau_2 \leftarrow 0$
$\forall z \in N$
    $M \leftarrow ((R{\downarrow}R_d z)(y) + (R{\uparrow}R_d z)(y))/2$
    $\tau_1 \leftarrow \tau_1 + M * d(y)$
    $\tau_2 \leftarrow \tau_2 + M$
output $\tau_1/\tau_2$
```

Fig. 4. The fuzzy-rough nearest neighbour algorithm - prediction

where ${\sigma_a}^2$ is the variance of attribute a, and $a_{\max}$ and $a_{\min}$ are the maximal and minimal occurring value of that attribute.

When using FRNN-FRS, the use of K is not required in principle: as $R(x,y)$ gets smaller, x tends to have only have a minor influence on $(R{\downarrow}C)(y)$ and $(R{\uparrow}C)(y)$. For FRNN-VQRS, this may generally not be true, because $R(x,y)$ appears in the numerator as well as the denominator of (12) and (13).

When dealing with real-valued decision features, the above algorithm can be modified to that found in Fig. 4. This is a zero order Takagi-Sugeno controller, with each neighbour acting as a rule. Here, the lower and upper approximations are defined as:

$$(R{\downarrow}R_d z)(x) = \inf_{y \in N} \mathcal{I}(R(x,y), R_d(y,z)) \tag{22}$$

$$(R{\uparrow}R_d z)(x) = \sup_{y \in N} \mathcal{T}(R(x,y), R_d(y,z)) \tag{23}$$

where R_d is the fuzzy tolerance relation for the decision feature d. In this paper, we use the same relation as that used for the conditional features. This need not be the case in general; indeed, it is conceivable that there may be situations where the use of a different similarity relation is sensible for the decision feature.

5 Experimentation

This section details the experimentation performed for the evaluation of the proposed algorithms for both classification and prediction tasks.

5.1 Classification

To demonstrate the power of the proposed fuzzy-rough NN approach, two sets of classification experiments were conducted. In the first set, the performance of the fuzzy and fuzzy-rough NN approaches were compared. The second set of experiments compared the proposed NN approaches (FRNN-FRS and FRNN-VQRS) with a variety of leading classification algorithms. Both sets of experiments were conducted over eight benchmark datasets from [4] and [22]. The details of the datasets used can be found in table 1. All of them have a crisp decision attribute.

Table 1. Dataset details

Dataset	Objects	Attributes
Cleveland	297	14
Glass	214	10
Heart	270	14
Letter	3114	17
Olitos	120	26
Water 2	390	39
Water 3	390	39
Wine	178	14

Fuzzy NN approaches. This section presents the initial experimental evaluation of the classification methods FNN, FRNN-O, FRNN-FRS and FRNN-VQRS for the task of pattern classification[2].

For FNN and FRNN-O, m is set to 2. For the new approaches, the fuzzy relation given in equation (21) was chosen. In the FRNN-FRS approach, we used the min t-norm and the Kleene-Dienes implicator I defined by $I(x, y) = \max(1-x, y)$. The FRNN-VQRS approach was implemented using $Q_l = Q_{(0.1,0.6)}$ and $Q_u = Q_{(0.2,1.0)}$, according to the general formula

$$Q_{(\alpha,\beta)}(x) = \begin{cases} 0, & x \leq \alpha \\ \frac{2(x-\alpha)^2}{(\beta-\alpha)^2}, & \alpha \leq x \leq \frac{\alpha+\beta}{2} \\ 1 - \frac{2(x-\beta)^2}{(\beta-\alpha)^2}, & \frac{\alpha+\beta}{2} \leq x \leq \beta \\ 1, & \beta \leq x \end{cases}$$

Initially, the impact of the number of neighbours K was investigated for the nearest neighbour approaches. K was initialized to $|\mathbb{U}|$, the number of objects in the training dataset, and then decremented by $1/30^{th}$ of $|\mathbb{U}|$ each time, resulting in 30 experiments for each dataset. For each choice of parameter K, $2\times$ 10-fold cross-validation was performed. The results of this for two datasets can be seen in Fig. 5 and Fig. 6.

It can be seen that FRNN-FRS is indeed unaffected by the choice of K for nominal-valued decision features. FRNN-O also appears to be relatively unaffected by K. For the Letter dataset, FRNN-VQRS and FNN exhibit degradation in classification performance as the number of nearest neighbours increases beyond 10. Therefore, for these methods the choice of K is an important consideration, with a value of around 10 neighbours being a sensible choice.

Based on this, further experimentation was conducted on a range of datasets. For this experimentation, each NN approach is run twice, the first time setting $K = 10$ and the second time with K set to the full set of training objects. Again, this is evaluated via $2\times$10-fold cross-validation.

The results of the experiments are shown in Table 2, where the average classification accuracy for the methods is recorded. For clarity, the method names have

[2] These methods and many more have been integrated into the WEKA package [29] and can be downloaded from: http://users.aber.ac.uk/rkj/book/programs.php

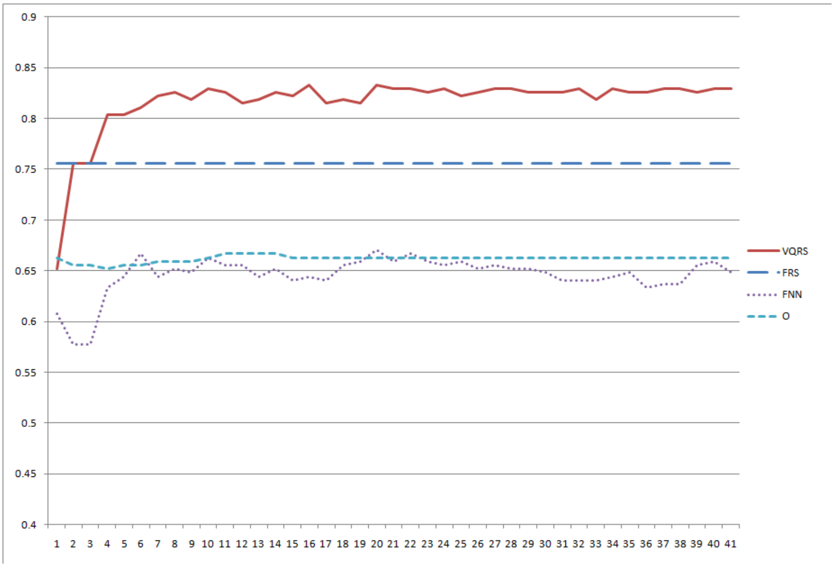

Fig. 5. Classification accuracy for the four methods and different values of K for the Heart dataset

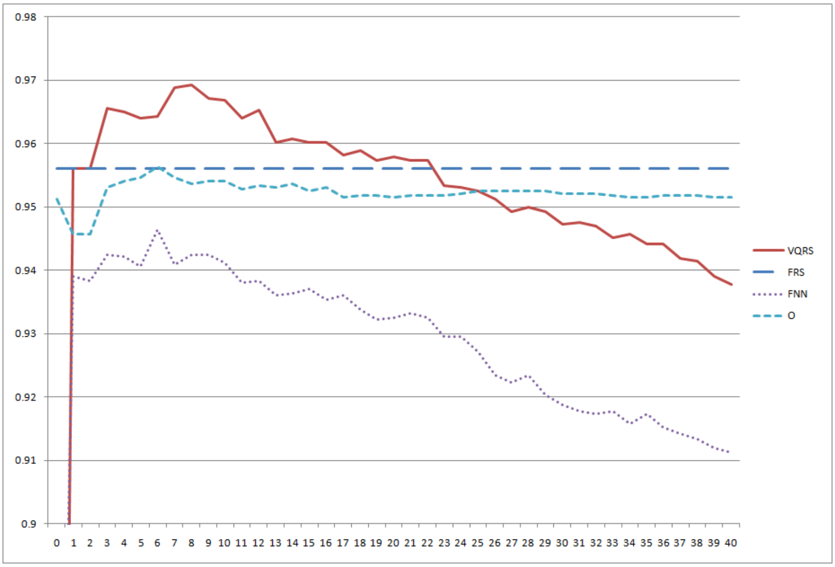

Fig. 6. Classification accuracy for the four methods and different values of K for the Letter dataset

Table 2. Nearest neighbour results

Dataset	FRS(10)	FRS	VQRS(10)	VQRS	FNN(10)	FNN	O(10)	O
Cleveland	53.21	53.21	59.41	53.89	50.19	53.89	47.50	47.50
Glass	73.13	73.13	69.36	38.06*	69.15	62.85*	71.22	71.22
Heart	76.30	76.30	82.04v	65.19*	66.11*	61.48*	66.48	66.30
Letter	95.76	95.76	96.69v	71.25*	94.25*	80.21*	95.45	95.26
Olitos	78.33	78.33	78.75	41.67*	63.75*	43.33*	65.83*	65.83*
Water 2	83.72	83.72	85.26	80.00	77.18*	80.00	79.62	79.62
Water 3	80.26	80.26	81.41	73.59*	74.49*	73.59*	73.08*	73.08*
Wine	98.02	98.02	97.75	63.79*	96.05	93.25*	95.78	95.78
Summary	(v/ /*)	(0/8/0)	(2/6/0)	(0/2/6)	(0/3/5)	(0/2/6)	(0/6/2)	(0/6/2)

been condensed in the table to: FRS (denoting FRNN-FRS), VQRS (denoting FRNN-VQRS), FNN (the standard fuzzy nearest neighbours algorithm), and O (denoting FRNN-O). A paired t-test was used to determine the statistical significance of the results at the 0.05 level when compared to FRNN-FRS. A 'v' next to a value indicates that the performance was statistically better than FRNN-FRS, and a '*' indicates that the performance was worse statistically. This is summarised by the final line in the table which shows the count of the number of statistically better, equivalent and worse results for each method in comparison to FRNN-FRS. For example (0/2/6) in the FNN column indicates that this method performed better than FRNN-FRS in zero datasets, equivalently to FRNN-FRS in two datasets, and worse than FRNN-FRS in six datasets.

For all datasets, either FRNN-FRS or FRNN-VQRS(10) yields the best results. Overall, FRNN-FRS produces the most consistent results. This is particularly remarkable considering the inherent simplicity of the method. FRNN-VQRS is best for `heart` and `letter`, which might be attributed to the comparative presence of noise in those datasets.

It is also interesting to consider the influence of the number of nearest neighbours. Both FRNN-FRS and FRNN-O remain relatively unaffected by changes in K. This could be explained in that, for FRNN-FRS, an infimum and supremum are used which can be thought of as a worst case and best case respectively. When more neighbours are considered, $R(x, y)$ values decrease as these neighbours are less similar, hence $I(R(x, y), C(x))$ increases, and $T(R(x, y), C(x))$ decreases. In other words, the more distant a neighbour is, the more unlikely it is to change the infimum and supremum value. For FRNN-O, again $R(x, y)$ decreases when more neighbours are added, and hence the value $R(x, y)C(x)$ that is added to the numerator is also small. Since each neighbour has the same weight in the denominator, the ratios stay approximately the same when adding new neighbours.

For FNN and FRNN-VQRS, increasing K can have a significant effect on classification accuracy. This is most clearly observed in the results for the `olitos` data, where there is a clear downward trend. For FRNN-VQRS, the ratio $|Ry \cap C|/|Ry|$ has to be calculated. Each neighbour has a different weight in the denominator, so the ratios can fluctuate considerably even when adding distant neighbours.

Table 3. Comparison results

Dataset	FRS	VQRS	IBk	JRip	PART	J48	SMO	NB
Cleveland	53.21	59.41	51.53	54.22	50.34	52.89	57.77	56.78
Glass	73.13	69.36	69.83	68.63	67.25	67.49	57.24*	49.99*
Heart	76.30	82.04v	76.11	80.93	74.26	78.52	84.07v	83.7v
Letter	95.76	96.69v	94.94	92.88*	93.82*	92.84*	89.05*	78.57*
Olitos	78.33	78.75	75.00	67.92*	63.33*	66.67*	87.5	76.67
Water 2	83.72	85.26	84.74	81.79	83.72	82.44	82.95	70.77*
Water 3	80.26	81.41	81.15	82.31	84.10	83.08	87.05v	85.51v
Wine	98.02	97.75	94.93	94.05	93.27	94.12	98.61	97.19
Summary	(v/ /*)	(2/6/0)	(0/8/0)	(0/6/2)	(0/6/2)	(0/6/2)	(2/4/2)	(2/3/3)

Comparison with leading approaches. In order to demonstrate the efficacy of the proposed methods, further experimentation was conducted involving several leading classifiers: IBk, JRip, PART, J48, SMO (a support vector-based method) and NB (naive bayes). The same datasets as above were used and 2×10-fold cross validation was performed. For FRNN-FRS and FRNN-VQRS, K was set to 10. The results can be seen in Table 3, with statistical comparisons again between each method and FRNN-FRS.

IBk [1] is a simple (non-fuzzy) K-nearest neighbour classifier that uses Euclidean distance to compute the closest neighbour (or neighbours if more than one object has the closest distance) in the training data, and outputs this object's decision as its prediction. JRip [5] learns propositional rules by repeatedly growing rules and pruning them. During the growth phase, features are added greedily until a termination condition is satisfied. Features are then pruned in the next phase subject to a pruning metric. Once the ruleset is generated, a further optimization is performed where classification rules are evaluated and deleted based on their performance on randomized data. PART [28,29] generates rules by means of repeatedly creating partial decision trees from data. The algorithm adopts a divide-and-conquer strategy such that it removes instances covered by the current ruleset during processing. Essentially, a classification rule is created by building a pruned tree for the current set of instances; the leaf with the highest coverage is promoted to a rule. J48 [20] creates decision trees by choosing the most informative features and recursively partitioning the data into subtables based on their values. Each node in the tree represents a feature with branches from a node representing the alternative values this feature can take according to the current subtable. Partitioning stops when all data items in the subtable have the same classification. A leaf node is then created, and this classification assigned. SMO [24] implements a sequential minimal optimization algorithm for training a support vector classifier. Pairwise classification is used to solve multi-class problems.

Both FRNN-FRS and FRNN-VQRS perform well. There are two datasets (`Water 3` and `Heart`) for which the methods are bettered by SMO and NB, but for the remainder their performance is equivalent to or better than all classifiers.

This is interesting, given the comparative algorithmic simplicity of FRNN-FRS and FRNN-VQRS.

5.2 Prediction

For the task of prediction, eight datasets were chosen that possess real-valued decision features (Table 4). The algae data sets[3] are provided by ERUDIT [11] and describe measurements of river samples for each of seven different species of alga, including river size, flow rate and chemical concentrations. The decision feature is the corresponding concentration of the particular alga. The housing dataset is taken from the Machine Learning Repository.

Seven methods were compared, namely the four nearest neighbour methods, IBk, SMOreg (support vector-based regression), LR (linear regression) and Pace. For the nearest neighbour methods, K was set to 10. Again, 2×10-fold cross validation was performed and the average root mean squared error (RMSE) was recorded.

Table 4. Dataset details

Dataset	Objects	Attributes
Algae A→G	187	11
Housing	506	13

The linear regression model [10] is applicable for numeric classification and prediction provided that the relationship between the input attributes and the output attribute is almost linear. The relation is then assumed to be a linear function of some parameters - the task being to estimate these parameters given training data. This is often accomplished by the method of least squares, which consists of finding the values that minimize the sum of squares of the residuals. Once the parameters are established, the function can be used to estimate the output values for unseen data. Projection adjustment by contribution estimation (Pace) regression [25] is a recent approach to fitting linear models, based on considering competing models. Pace regression improves on classical ordinary least squares regression by evaluating the effect of each variable and using a clustering analysis to improve the statistical basis for estimating their contribution to the overall regression. SMOreg is a sequential minimal optimization algorithm for training a support vector regression using polynomial or Radial Basis Function kernels [19,24]. It reduces support vector machine training down to a series of smaller quadratic programming subproblems that have an analytical solution. This has been shown to be very efficient for prediction problems using linear support vector machines and/or sparse data sets.

The results for the prediction experimentation can be seen in Table 5. It can be seen that FRNN-O and IBk perform poorly, and the other methods perform similarly to FRNN-FRS. The average RMSEs for FRNN-FRS and FRNN-VQRS are generally lower than those obtained for the other algorithms.

[3] See http://archive.ics.uci.edu/ml/datasets/Coil+1999+Competition+Data

Table 5. Prediction results (RMSE)

Dataset	FRS	VQRS	FNN	O	IBk	SMOreg	LR	Pace
Algae A	17.15	16.81	15.79	24.55*	24.28*	17.97	18.00	18.18
Algae B	10.77	10.57	10.68	13.04*	17.18*	10.08	10.30	10.06
Algae C	6.81	6.68	6.99	8.16*	9.07*	7.12	7.11	7.26
Algae D	2.91	2.88	3.04	3.47*	4.62*	2.99	3.86	3.95
Algae E	6.88	6.85	7.38	9.10*	9.02*	7.18	7.61	7.59
Algae F	10.40	10.33	11.24	12.60*	13.51*	10.09	10.33	9.65
Algae G	4.97	4.84	5.23	5.38	6.48	4.96	5.21	4.96
Housing	4.72	4.85	6.62*	24.27*	4.59	4.95	4.80	4.79
Summary	(v/ /*)	(0/8/0)	(0/7/1)	(0/1/7)	(0/2/6)	(0/8/0)	(0/8/0)	(0/8/0)

6 Conclusion and Future Work

This paper has presented two new techniques for fuzzy-rough classification based on the use of lower and upper approximations w.r.t. fuzzy tolerance relations. The difference between them is in the definition of the approximations: while FRNN-FRS uses "traditional" operations based on a t-norm and an implicator, FRNN-VQRS uses a fuzzy quantifier-based approach. The results show that these methods are effective, and that they are very competitive with existing methods for both classification and prediction. Further investigation is still needed to adequately explain the impact of the choice of fuzzy relations, connectives and quantifiers. Of particular importance is the choice of relation composition operator as this determines the overall similarity of objects based on the full set of data features. The use of a t-norm for this operation is sensible from a theoretical viewpoint, but may introduce problems from a practical perspective as the overall similarity of a pair of objects will be zero if these objects have zero similarity for just one of their features. Therefore, an alternative method of combining relations is desirable.

Also, the impact of a feature selection preprocessing step upon classification accuracy needs to be investigated. It is expected that feature selectors that incorporate fuzzy relations expressing closeness of objects (see e.g. [8,15]) should be able to further improve the effectiveness of the classification methods presented here.

Acknowledgment. Chris Cornelis would like to thank the Research Foundation—Flanders for funding his research.

References

1. Aha, D.: Instance-based learning algorithm. Machine Learning 6, 37–66 (1991)
2. Bhatt, R.B., Gopal, M.: FRID: Fuzzy-Rough Interactive Dichotomizers. In: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2004), pp. 1337–1342 (2004)

3. Bian, H., Mazlack, L.: Fuzzy-Rough Nearest-Neighbor Classification Approach. In: Proceeding of the 22nd International Conference of the North American Fuzzy Information Processing Society (NAFIPS), pp. 500–505 (2003)
4. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. University of California, Irvine (1998), http://archive.ics.uci.edu/ml/
5. Cohen, W.W.: Fast effective rule induction. In: Machine Learning: Proceedings of the 12th International Conference, pp. 115–123 (1995)
6. Cornelis, C., De Cock, M., Radzikowska, A.M.: Vaguely Quantified Rough Sets. In: An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.) RSFDGrC 2007. LNCS (LNAI), vol. 4482, pp. 87–94. Springer, Heidelberg (2007)
7. Cornelis, C., De Cock, M., Radzikowska, A.M.: Fuzzy Rough Sets: from Theory into Practice. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Handbook of Granular Computing. Wiley, Chichester (2008)
8. Cornelis, C., Hurtado Martín, G., Jensen, R., Slezak, D.: Feature Selection with fuzzy decision reducts. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 284–291. Springer, Heidelberg (2008)
9. Duda, R., Hart, P.: Pattern Classification and Scene Analysis. Wiley, New York (1973)
10. Edwards, A.L.: An Introduction to Linear Regression and Correlation. W.H. Freeman, San Francisco (1976)
11. European Network for Fuzzy Logic and Uncertainty Modelling in Information Technology (ERUDIT), Protecting rivers and streams by monitoring chemical concentrations and algae communities, Computational Intelligence and Learning (CoIL) Competition (1999)
12. Greco, S., Inuiguchi, M., Slowinski, R.: Fuzzy rough sets and multiple-premise gradual decision rules. International Journal of Approximate Reasoning 41, 179–211 (2005)
13. Hong, T.P., Liou, Y.L., Wang, S.L.: Learning with Hierarchical Quantitative Attributes by Fuzzy Rough Sets. In: Proceedings of the Joint Conference on Information Sciences, Advances in Intelligent Systems Research(2006)
14. Hsieh, N.-C.: Rule Extraction with Rough-Fuzzy Hybridization Method. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 890–895. Springer, Heidelberg (2008)
15. Jensen, R., Shen, Q.: Fuzzy-Rough Sets Assisted Attribute Selection. IEEE Transactions on Fuzzy Systems 15(1), 73–89 (2007)
16. Jensen, R., Shen, Q.: Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches. Wiley-IEEE Press (2008)
17. Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy K-nearest neighbor algorithm. IEEE Trans. Systems Man Cybernet. 15(4), 580–585 (1985)
18. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishing, Dordrecht (1991)
19. Platt, J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: Schölkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods - Support Vector Learning. MIT Press, Cambridge (1998)
20. Quinlan, J.R.: C4.5: Programs for Machine Learning. The Morgan Kaufmann Series in Machine Learning. Morgan Kaufmann Publishers, San Mateo (1993)
21. Radzikowska, A.M., Kerre, E.E.: A comparative study of fuzzy rough sets. Fuzzy Sets and Systems 126(2), 137–155 (2002)
22. Sarkar, M.: Fuzzy-Rough nearest neighbors algorithm. Fuzzy Sets and Systems 158, 2123–2152 (2007)

23. Shen, Q., Chouchoulas, A.: A rough-fuzzy approach for generating classification rules. Pattern Recognition 35(11), 2425–2438 (2002)
24. Smola, A.J., Schölkopf, B.: A Tutorial on Support Vector Regression, NeuroCOLT2 Technical Report Series - NC2-TR-1998-030 (1998)
25. Wang, Y.: A new approach to fitting linear models in high dimensional spaces, PhD Thesis, Department of Computer Science, University of Waikato (2000)
26. Wang, X., Yang, J., Teng, X., Peng, N.: Fuzzy-Rough Set Based Nearest Neighbor Clustering Classification Algorithm. In: Wang, L., Jin, Y. (eds.) FSKD 2005. LNCS (LNAI), vol. 3613, pp. 370–373. Springer, Heidelberg (2005)
27. Wang, X., Tsang, E.C.C., Zhao, S., Chen, D., Yeung, D.S.: Learning fuzzy rules from fuzzy samples based on rough set technique. Information Sciences 177(20), 4493–4514 (2007)
28. Witten, I.H., Frank, E.: Generating Accurate Rule Sets Without Global Optimization. In: Proceedings of the 15th International Conference on Machine Learning. Morgan Kaufmann Publishers, San Francisco (1998)
29. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, San Francisco (2000)
30. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)