Fuzzy Rough Nearest Neighbour Classification and Prediction

Richard Jensen[a], Chris Cornelis[b]

[a]*Dept. of Comp. Sci., Aberystwyth University, Ceredigion, SY23 3DB, Wales, UK*
[b]*Dept. of Appl. Math. and Comp. Sci., Ghent University, Gent, Belgium*

Abstract

In this paper, we propose a nearest neighbour algorithm that uses the lower and upper approximations from fuzzy rough set theory in order to classify test objects, or predict their decision value. It is shown experimentally that our method outperforms other nearest neighbour approaches (classical, fuzzy and fuzzy-rough ones) and that it is competitive with leading classification and prediction methods. Moreover, we show that the robustness of our methods against noise can be enhanced effectively by invoking the approximations of the Vaguely Quantified Rough Set (VQRS) model.

Keywords: fuzzy rough sets, classification, prediction, nearest neighbours

1. Introduction

Fuzzy sets [42] and rough sets [28] address two important, complementary characteristics of imperfect data and knowledge: the former model vague information by expressing that objects belong to a set or relation to a given degree, while the latter provide approximations of concepts in the presence of incomplete information. A hybrid fuzzy rough set model was first proposed by Dubois and Prade in [12], was later extended and/or modified by many authors, and has been applied successfully in various domains, most notably machine learning.

The K-nearest neighbour (KNN) algorithm [13] 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 [24]. 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 [33], the FNN algorithm has problems dealing adequately with insufficient knowledge. To address this problem, he introduced a so-called fuzzy-rough ownership function. However, this method (called FRNN-O throughout this paper) does not refer to the main ingredients of rough set theory, i.e., the lower and upper approximation.

In this paper, therefore, we propose a nearest neighbour algorithm based on fuzzy-rough lower and upper approximations. We consider two variants of this algorithm: one is based on the common implicator/t-norm based branch of fuzzy rough sets introduced by Radzikowska and Kerre [32], while the other uses the more recent Vaguely Quantified Rough Set (VQRS) model from [10]. The discerning feature of the VQRS approach is the introduction of vague quantifiers like 'some' or 'most' into the approximations, which according to [10] makes the model more robust in the presence of classification errors. In this paper, we take up this claim by evaluating VQRS's noise-handling potential in the context of classification and prediction.

The remainder of this paper is structured as follows: Section 2 provides the necessary background details for fuzzy rough set theory, while Section 3 and 4 are concerned with the fuzzy NN approach, and Sarkar's fuzzy-rough ownership function, respectively. Section 5 outlines our algorithm, while comparative experimentation on a series of classification and prediction problems is provided in Section 6, both with and without noise. The paper is concluded in section 7. Finally, let us mention that a preliminary version of some of the ideas developed in this paper appears in the conference paper [20].

2. Hybridization of Rough Sets and Fuzzy Sets

2.1. Rough Set Theory

Rough set theory (RST) [29] 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 $(X, \mathbb{A})$ be an information system, where X 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 $a : X \to V_a$ 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:

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

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

$$R_B{\downarrow}A = \{x \in X \mid [x]_B \subseteq A\} \tag{2}$$
$$R_B{\uparrow}A = \{x \in X \mid [x]_B \cap A \neq \emptyset\} \tag{3}$$

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

A *decision system* $(X, \mathbb{A} \cup \{d\})$ is a special kind of information system, used in the context of classification or prediction, in which d ($d \notin \mathbb{A}$) is a designated attribute called the decision attribute. In case d is nominal (i.e., in a classification problem), the equivalence classes $[x]_d$ are called decision classes; the set of decision classes is denoted $\mathcal{C}$ in this paper.

2.2. Fuzzy Set Theory

Fuzzy set theory [42] 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

$$Ry(x) = R(x, y) \tag{4}$$

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

$$R(x, x) = 1 \tag{5}$$
$$R(x, y) = R(y, x) \tag{6}$$

hold for all x and y in X, then R is called a fuzzy tolerance relation.

If X is finite, the cardinality of A is calculated by

$$|A| = \sum_{x \in X} A(x). \tag{7}$$

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 $[0,1]^2 \to [0,1]$ mapping satisfying $\mathcal{T}(1,x) = x$, for all x in $[0,1]$. In this paper, we use $\mathcal{T}_M$ defined by $\mathcal{T}_M(x,y) = \min(x,y)$, for x, y in $[0,1]$. On the other hand, an implicator is any $[0,1]^2 \to [0,1]$-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. In this paper, we use $\mathcal{I}_M$ defined by $\mathcal{I}_M(x,y) = \max(1-x,y)$ (Kleene-Dienes implicator) for x, y in $[0,1]$.

2.3. Fuzzy Rough Set Theory

Research on the hybridization of fuzzy sets and rough sets emerged in the late 1980s [12] and has flourished recently (e.g. [10, 21, 22]). It has focused predominantly on fuzzifying the formulas for the lower and upper approximations. In doing so, the following two guiding principles have been widely adopted:

- The set A may be generalized to a fuzzy set in X, allowing that objects can belong to a given concept to varying degrees.
- Rather than assessing objects' indiscernibility, we may measure their *approximate equality*. As a result, objects are categorized into classes, or granules, with "soft" boundaries based on their similarity to one another. As such, abrupt transitions between classes are replaced by gradual ones, allowing that an element can belong (to varying degrees) to more than one class.

More formally, the approximate equality between objects with continuous attribute values is modelled by means of a fuzzy relation R in X 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.

Given a fuzzy tolerance relation R and a fuzzy set A in X, the lower and upper approximation of A by R can be constructed in several ways. A general definition [32] is the following:

$$(R{\downarrow}A)(x) = \inf_{y \in X} \mathcal{I}(R(x,y), A(y)) \tag{8}$$

$$(R{\uparrow}A)(x) = \sup_{y \in X} \mathcal{T}(R(x,y), A(y)) \tag{9}$$

Here, $\mathcal{I}$ is an implicator and $\mathcal{T}$ a t-norm. When A is a crisp (classical) set and R is an equivalence relation in X, the traditional lower and upper approximation are recovered. While this is often perceived as an advantage, it also brings along some problems. In particular, the use of the inf and sup operations makes (8) and (9) subject to noise just like the universal and existential quantifier $\forall$ and $\exists$ do in the crisp case.

For this reason, the concept of vaguely quantified rough sets was introduced in [10]. 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

$$(R{\downarrow}^{Q_u}A)(y) = Q_u\left(\frac{|Ry \cap A|}{|Ry|}\right) = Q_u\left(\frac{\sum_{x\in X}\min(R(x,y),A(x))}{\sum_{x\in X}R(x,y)}\right) \quad (10)$$

$$(R{\uparrow}^{Q_l}A)(y) = Q_l\left(\frac{|Ry \cap A|}{|Ry|}\right) = Q_l\left(\frac{\sum_{x\in X}\min(R(x,y),A(x))}{\sum_{x\in X}R(x,y)}\right) \quad (11)$$

where the fuzzy set intersection is defined by the min t-norm.

Examples of fuzzy quantifiers can be generated by means of the following parametrized formula, for $0 \leq \alpha < \beta \leq 1$, and x in $[0,1]$,

$$Q_{(\alpha,\beta)}(x) = \begin{cases} 0, & x \leq \alpha \\ \frac{2(x-\alpha)^2}{(\beta-\alpha)^2}, & \alpha \leq x \leq \frac{\alpha+\beta}{2} \\ 1 - \frac{2(x-\beta)^2}{(\beta-\alpha)^2}, & \frac{\alpha+\beta}{2} \leq x \leq \beta \\ 1, & \beta \leq x \end{cases} \quad (12)$$

In this paper, $Q_{(0.1,0.6)}$ and $Q_{(0.2,1)}$ are used respectively to reflect the vague quantifiers *some* and *most* from natural language. As an important difference to (8) and (9), 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. In this case, note also that when

$$Q_{>x_l}(x) = \begin{cases} 0, & x \leq x_l \\ 1, & x > x_l \end{cases} \qquad Q_{\geq x_u}(x) = \begin{cases} 0, & x < x_u \\ 1, & x \geq x_u \end{cases}$$

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

with $0 \leq x_l < x_u \leq 1$ are used as quantifiers, we recover Ziarko's variable precision rough set model [45, 47], and moreover when we use

$$Q_{\exists}(x) = \begin{cases} 0, & x = 0 \\ 1, & x > 0 \end{cases} \qquad Q_{\forall}(x) = \begin{cases} 0, & x < 1 \\ 1, & x = 1 \end{cases}$$

we obtain Pawlak's standard rough set model as a particular case of the VQRS approach, assuming that R is a crisp equivalence relation.

As such, the VQRS model puts dealing with noisy data into an interesting new perspective: it inherits both the flexibility of VPRS for dealing with classification errors (by relaxing the membership conditions for the lower approximation, and tightening those for the upper approximation) and that of fuzzy sets for expressing partial constraint satisfaction (by distinguishing different levels of membership to the upper/lower approximation). This model has been employed for feature selection in [8].

Another approach that blurs the distinction between rough and fuzzy sets has been proposed in [30]. The research was fueled by the concern that a purely numeric fuzzy set representation may be too precise; a concept is described exactly once its membership function has been defined (a similar motivation to that of Type-2 fuzzy sets). This seems as though excessive precision is required in order to describe imprecise concepts. The solution proposed is termed a shadowed set, which itself does not use exact membership values but instead employs basic truth values and a zone of uncertainty (the unit interval). A shadowed set could be thought of as an approximation of a fuzzy set or family of fuzzy sets where elements may belong to the set with certainty (membership of 1), possibility (unit interval) or not at all (membership of 0). This can be seen to be analogous to the definitions of the rough set regions: the positive region (certainty), the boundary region (possibility) and the negative region (no membership).

Given a fuzzy set, a shadowed set can be induced by elevating those membership values around 1 and reducing membership values around 0 until a certain threshold level is achieved. Any elements that do not belong to the set with a membership of 1 or 0 are assigned a unit interval, [0,1], considered to be a non-numeric model of membership grade. These regions of uncertainty are referred to as *shadows*. In fuzzy set theory, vagueness is distributed across the entire universe of discourse, but in shadowed sets this vagueness is localized in the shadow regions. As with fuzzy sets, the basic set operations (union, intersection and complement) can be defined for shadowed sets, as well as shadowed relations.

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 [19, 34] but mainly ignores the direct use of fuzzy-rough concepts.

The induction of gradual decision rules, based on fuzzy-rough hybridization, is given in [16]. 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 [38], 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 [4] where a method for fuzzy decision tree construction is given that employs the fuzzy-rough ownership function discussed in Section 4. 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 [21], 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 [18] for generating certain and possible rulesets from hierarchical data.

3. Fuzzy Nearest Neighbour Classification

The fuzzy K-nearest neighbour (FNN) algorithm [24] 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:

$$C'(y) = \sum_{x \in N} R(x, y) C(x) \tag{13}$$

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:

$$R(x, y) = \frac{||y - x||^{-2/(m-1)}}{\sum_{j \in N} ||y - j||^{-2/(m-1)}} \tag{14}$$

where $|| \cdot ||$ denotes the Euclidean norm, and m is a parameter that controls the overall weighting of the similarity. In this paper, m is set to the default value 2. Assuming crisp classes, Algorithm 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 $O(|X| + K \cdot |\mathcal{C}|)$.

Algorithm 1: The FNN algorithm

Input: X, the training data; $\mathcal{C}$, the set of decision classes; y, the object to be classified; K, the number of nearest neighbours
Output: Classification for y
begin
  $N \leftarrow$ getNearestNeighbours(y,K)
  foreach $C \in \mathcal{C}$ **do**
    $C'(y) = \sum_{x \in N} R(x, y)C(x)$
  end
  output $\arg\max_{C \in \mathcal{C}} (C'(y))$
end

4. Fuzzy-rough Ownership

Initial attempts to combine the FNN algorithm with concepts from fuzzy rough set theory were presented in [33, 37] and improved in [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,

$$\tau_C(y) = \frac{\sum\limits_{x \in X} R(x,y)C(x)}{|X|} \tag{15}$$

In this, the fuzzy relation R is determined by:

$$R(x,y) = \exp\left(-\sum_{a \in \mathbb{A}} \kappa_a (a(y) - a(x))^{2/(m-1)}\right) \tag{16}$$

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

$$\kappa_a = \frac{|X|}{2\sum\limits_{x \in X} ||a(y) - a(x)||^{2/(m-1)}} \tag{17}$$

$\tau_C(y)$ 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 Algorithm 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 (15). Finally, when all training objects have been considered, the algorithm outputs the class with highest membership. The algorithm's complexity is $O(|\mathbb{A}|.|X| + |X| \cdot (|\mathbb{A}| + |\mathcal{C}|))$.

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 [5]. However, the authors did not state how to use the upper and lower approximations to derive classifications. Also, in [2], a rough-fuzzy weighted K-nearest leader classifier was proposed; however, the concepts of lower and upper approximations were redefined for this purpose and have no overlap with the traditional definitions.

Algorithm 2: The fuzzy-rough ownership nearest neighbour algorithm

Input: X, the training data; $\mathbb{A}$, the set of conditional features; $\mathcal{C}$, the set of decision classes;y, the object to be classified.

Output: Classification for y

begin

  foreach $a \in \mathbb{A}$ **do**

    $\kappa_a = |X|/2 \sum_{x \in X} ||a(y) - a(x)||^{2/(m-1)}$

  end

  $N \leftarrow |X|$

  foreach $C \in \mathcal{C}$ **do** $\tau_C(y) = 0$

  foreach $x \in N$ **do**

    $d = \sum_{a \in \mathbb{A}} \kappa_a (a(y) - a(x))^2$

    foreach $C \in \mathcal{C}$ **do**

      $\tau_C(y) += \frac{C(x) \cdot \exp(-d^{1/(m-1)})}{|N|}$

    end

  end

  output $\arg\max_{C \in \mathcal{C}} \tau_C(y)$

end

5. Fuzzy-Rough Nearest Neighbours

In this section, we propose a fuzzy-rough nearest neighbours (FRNN) algorithm where the nearest neighbours are used to construct the fuzzy lower and upper approximations of decision classes, and test instances are classified based on their membership to these approximations. The algorithm, combining fuzzy-rough approximations with the ideas of the classical FNN approach, can be seen in Algorithm 3.

The algorithm is dependent on the choice of a fuzzy tolerance relation R. In this paper, we construct R as follows: given the set of conditional attributes $\mathbb{A}$, R is defined by

$$R(x, y) = \min_{a \in \mathbb{A}} R_a(x, y) \tag{18}$$

in which $R_a(x, y)$ is the degree to which objects x and y are similar for attribute a. Many options are possible, here we choose

$$R_a(x, y) = 1 - \frac{|a(x) - a(y)|}{|a_{\max} - a_{\min}|} \tag{19}$$

Algorithm 3: The fuzzy-rough nearest neighbour algorithm

Input: X, the training data; $\mathcal{C}$, the set of decision classes; y, the object to be classified
Output: Classification for y
begin
  $N \leftarrow$ getNearestNeighbours(y,K)
  $\tau \leftarrow 0$, $Class \leftarrow \emptyset$
  foreach $C \in \mathcal{C}$ **do**
    if $((R{\downarrow}C)(y) + (R{\uparrow}C)(y))/2 \geq \tau$ **then**
      $Class \leftarrow C$
      $\tau \leftarrow ((R{\downarrow}C)(y) + (R{\uparrow}C)(y))/2$
    end
  end
  output *Class*
end

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.

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 of y's neighbours belong to C, while a high value of $(R{\uparrow}C)(y)$ means that at least one neighbour belongs to that class. 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 $O(|\mathcal{C}| \cdot (2|X|))$.

When dealing with real-valued decision features, the above algorithm can be modified to that found in Algorithm 4. This can be interpreted as a zero order Takagi-Sugeno controller [36], with each neighbour acting as a rule, and the average of the test object's membership to the lower and upper approximation as the activation degree. 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. Line (10) of the algorithm is only meant to make sure that the algorithm returns a prediction under all circumstances. Note that, with $\mathcal{I} = \mathcal{I}_M$ and $\mathcal{T} = \mathcal{T}_M$, condition $\tau_2 = 0$ is only fulfilled when $R(y, z) = 1$ for all neighbours z in N (total similarity of the test object and the nearest neighbours), but $R_d(z_1, z_2) = 0$ for every z_1, z_2 in N (total dissimilarity between any two neighbours' decision values).

Algorithm 4: The fuzzy-rough nearest neighbour algorithm - prediction

Input: X, the training data; d, the decision feature; y, the object for which to find a prediction
Output: Classification for y
begin
  $N \leftarrow$ getNearestNeighbours(y,K)
  $\tau_1 \leftarrow 0$, $\tau_2 \leftarrow 0$
  foreach $z \in N$ **do**
    $M \leftarrow ((R{\downarrow}R_d z)(y) + (R{\uparrow}R_d z)(y))/2$
    $\tau_1 \leftarrow \tau_1 + M * d(z)$
    $\tau_2 \leftarrow \tau_2 + M$
  end
  if $\tau_2 > 0$ **then**
    output τ_1/τ_2
  else
    output $\sum_{z \in N} d(z)/|N|$
  end
end

By its reliance on the approximations of standard fuzzy rough set theory, the algorithms presented above may be impacted by noise. This is due to the use of sup and inf to generalize the existential and universal quantfier, respectively. A change in a single object can result in drastic changes to the lower and upper approximations, accordingly. Another (related) problem with the approach is that, for classification, it is not affected by the choice of K; indeed, it may be verified that in the case of crisp decisions (Algorithm

3), only the single nearest neighbour is used for classification.[2] Although this can be seen as beneficial with regard to the problem of parameter selection, in reality it means that its classification decisions are based on a single object only, making the approach even more succeptible to noisy data.

For this reason, we also propose VQNN (Vaguely Quantified Nearest Neighbours), a variant of FRNN in which $R{\downarrow}C$ and $R{\uparrow}C$ are replaced by $R{\downarrow}^{Q_u}C$ and $R{\uparrow}^{Q_l}C$, respectively. Analogously, VQNN2 is a variant of FRNN2 in which $R{\downarrow}R_dz$ and $R{\uparrow}R_dz$ are replaced by $R{\downarrow}^{Q_u}R_dz$ and $R{\uparrow}^{Q_l}R_dz$, respectively.

As we have already mentioned, for FRNN, the use of K is of no importance. For FRNN2, its impact is very limited, since 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 VQNN and VQNN2, this may generally not be true, because $R(x, y)$ appears in the numerator as well as the denominator of (10) and (11).

6. Experimentation

To demonstrate the power of the proposed approach, several sets of experiments were conducted. In the first set, the impact of K, the number of nearest neighbours was investigated for of the fuzzy and fuzzy-rough approaches discussed in Section 3, 4 and 5. In the second set, a comparative investigation was undertaken to compare the classification performance of these methods. The third set of experiments compares FRNN and VQNN with a variety of leading classification algorithms. The fourth set investigates the applicability of the proposed methods to the task of prediction, comparing it to a number of leading prediction algorithms. The final set of experiments investigates how well VQNN handles a range of noise levels introduced to the benchmark data.

The experiments were conducted over 16 benchmark datasets (8 for classification and 8 for prediction, depending on the decision attribute). The details of the datasets used can be found in table 1. The `Algae` datasets[3] are provided by ERUDIT [15] and describe measurements of river samples for each of seven different species of alga, including river size, flow rate and

[2] This assumes that there is exactly one nearest neighbour z such that $R(z, y)$ is maximal among all neighbours.

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

Table 1: Dataset details

Dataset	Objects	Attributes	Decision
Cleveland	297	14	nominal
Glass	214	10	nominal
Heart	270	14	nominal
Letter	3114	17	nominal
Olitos	120	26	nominal
Water 2	390	39	nominal
Water 3	390	39	nominal
Wine	178	14	nominal
Algae A→G	187	11	continuous
Housing	506	13	continuous

chemical concentrations. The decision feature is the corresponding concentration of the particular alga. The Letter dataset comes from [33], while the other datasets are taken from the Machine Learning Repository [6].

The fuzzy-rough approaches discussed in this paper, along with many more, have been integrated into the WEKA package [41] and can be downloaded from: http://users.aber.ac.uk/rkj/book/programs.php.

6.1. Impact of K

Initially, the impact of the number of neighbours K on classification accuracy was investigated for the nearest neighbour approaches. Here, 41 experiments were conducted ($K = 1, \ldots, 41$) for each dataset. For each choice of parameter K, 2×10-fold cross-validation was performed. The results can be seen in Figs. 1 to 4.

The experiments confirm that, for classification, FRNN is insensitive to the value of parameter K, as is FRNN-O to a lesser extent. FNN and VQNN, on the other hand, are affected more substantially by K. This is most clearly observed in the results for the Glass and Letter data, where there is a clear downward trend. In general for VQNN, a choice of K in the range 5 to 10 appears to produce the best results. The trend for VQNN seems to be an increase in accuracy in this range followed by a steady drop as K increases further. This is to be expected as there is benefit in considering a number of neighbours to reduce the effect of noise, but as more neighbours

are considered the distinction between classes becomes less clear.

6.2. Comparative study of NN Approaches

This section presents the experimental evaluation of the classification methods FNN, FRNN-O, FRNN and VQNN for the task of classification. For this experimentation, in accordance with the findings from the previous paragraph, FRNN and FRNN-O are run with K set to the full set of training objects, while for VQNN and FNN $K = 10$ is used. Again, this is evaluated via 2×10-fold cross-validation.

The results of the experiments are shown in Table 2, where the average classification accuracy for the methods is recorded. A paired t-test was used to determine the statistical significance of the results at the 0.05 level when compared to FRNN. A 'v' next to a value indicates that the performance was statistically better than FRNN, 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. For example (0/3/5) in the FNN column indicates that this method performed better than FRNN in zero datasets, equivalently to FRNN in three datasets, and worse than FRNN in five datasets.

For all datasets, either FRNN or VQNN yields the best results. VQNN is best for `Heart` and `Letter`, which might be attributed to the comparative presence of noise in those datasets.

Table 2: Nearest neighbour classification results (accuracy)

Dataset	FRNN	VQNN	FNN	FRNN-O
`Cleveland`	53.21	59.41	50.19	47.50
`Glass`	73.13	69.36	69.15	71.22
`Heart`	76.30	82.04v	66.11*	66.30
`Letter`	95.76	96.69v	94.25*	95.26
`Olitos`	78.33	78.75	63.75*	65.83*
`Water 2`	83.72	85.26	77.18*	79.62
`Water 3`	80.26	81.41	74.49*	73.08*
`Wine`	98.02	97.75	96.05	95.78
Summary	(v/ /*)	(2/6/0)	(0/3/5)	(0/6/2)

6.3. Comparison with Other Classification Methods

In order to demonstrate the efficacy of the proposed methods, further experimentation was conducted involving several leading classifiers. 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 [7] 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 [40, 41] 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 [31] 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 [35] implements a sequential minimal optimization algorithm for training a support vector classifier. Pairwise classification is used to solve multi-class problems. Finally, NB (Naive Bayes) is a simple probabilistic classifier based on applying Bayes' theorem with strong independence assumptions.

The same datasets as above were used and 2×10-fold cross validation was performed. The results can be seen in Table 3, with statistical comparisons again between each method and FRNN. There are two datasets (`Water 3` and `Heart`) for which FRNN is bettered by SMO and NB, but for the remainder its performance is equivalent to or better than all classifiers.

6.4. Prediction

For the task of prediction, we compared FRNN and VQNN ($K = 10$) to IBk, and three other prediction approaches from the literature. SMOreg is a sequential minimal optimization algorithm for training a support vector

Table 3: Comparison of FRNN with leading classifiers (accuracy)

Dataset	FRNN	IBk	JRip	PART	J48	SMO	NB
Cleveland	53.21	51.53	54.22	50.34	52.89	57.77	56.78
Glass	73.13	69.83	68.63	67.25	67.49	57.24*	49.99*
Heart	76.30	76.11	80.93	74.26	78.52	84.07v	83.70v
Letter	95.76	94.94	92.88*	93.82*	92.84*	89.05*	78.57*
Olitos	78.33	75.00	67.92*	63.33*	66.67*	87.5	76.67
Water 2	83.72	84.74	81.79	83.72	82.44	82.95	70.77*
Water 3	80.26	81.15	82.31	84.10	83.08	87.05v	85.51v
Wine	98.02	94.93	94.05	93.27	94.12	98.61	97.19
Summary	(v/ /*)	(0/8/0)	(0/6/2)	(0/6/2)	(0/6/2)	(2/4/2)	(2/3/3)

regression using polynomial or Radial Basis Function kernels [35]. 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 linear regression (LR) model [14] 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 [39] 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.

Again, 2×10-fold cross validation was performed and this time the average root mean squared error (RMSE) was recorded. The results for the prediction experiment can be seen in Table 4. It can be seen that all methods perform similarly to FRNN and VQNN. The average RMSEs for FRNN and VQNN

are generally better than those obtained for the other algorithms.

Table 4: Prediction results (RMSE)

Dataset	FRNN	VQNN	IBk	SMOreg	LR	Pace
`Algae A`	17.15	16.81	24.28*	17.97	18.00	18.18
`Algae B`	10.77	10.57	17.18*	10.08	10.30	10.06
`Algae C`	6.81	6.68	9.07*	7.12	7.11	7.26
`Algae D`	2.91	2.88	4.62*	2.99	3.86	3.95
`Algae E`	6.88	6.85	9.02*	7.18	7.61	7.59
`Algae F`	10.40	10.33	13.51*	10.09	10.33	9.65
`Algae G`	4.97	4.84	6.48	4.96	5.21	4.96
`Housing`	4.72	4.85	4.59	4.95	4.80	4.79
Summary	(v/ /*)	(0/8/0)	(0/7/1)	(0/8/0)	(0/8/0)	(0/8/0)

6.5. Noise Investigation

The final set of experiments investigates the impact on the classification algorithms of noise. For this purpose, different levels of artificial class noise were added to the benchmark datasets, i.e., class memberships of selected objects were randomly changed. The noise levels are given as a percentage, e.g., if the noise level is 10% this denotes that 10% of the data has noise applied, the rest remain unchanged. In this experiment, 10×10-fold cross validation is performed for each noise level for each algorithm.

Tables 5 and 6 show the results of this experimentation. In the first table, the number of datasets is given for which VQNN is better statistically than the specified method. In the second table, the number of datasets is given for which VQNN is statistically worse. It can be seen that as the amount of noise increases, VQNN performs increasingly better than FRNN demonstrating its better noise-handling approach. This is also the case when compared to IBk, J48 and Part. VQNN performs well against JRip across noise levels. It performs comparably with NB and SMO until extreme noise levels are reached (60% and 80% noise). At this point, it appears to be the case that there is too much noise for VQNN to cope with; the poorer performance probably being due to the nearest neighbour approach itself. The totals given in the tables show that VQNN reaches its peak in noise

tolerance at the 25% level, when compared to the other methods it performs statistically better in 34 out of 56 experiments, and statistically worse in only 2 of them.

Table 5: Number of datasets in which VQNN performs statistically better than other classification methods, for increasing noise levels

Method	0%	5%	10%	15%	20%	25%	40%	60%	80%
FRNN	3	5	6	6	6	7	7	9	7
SMO	2	1	2	1	1	1	1	0	0
IBk	4	4	6	6	8	9	9	9	7
J48	1	3	4	6	7	7	5	5	5
JRip	1	3	2	2	3	3	3	3	3
Part	3	4	5	5	5	5	5	5	4
NB	3	3	3	3	2	2	2	1	1
Total	17	23	28	29	32	34	32	32	27

Table 6: Number of datasets in which VQNN performs statistically worse than other classification methods, for increasing noise levels

Method	0%	5%	10%	15%	20%	25%	40%	60%	80%
FRNN	0	0	0	0	0	0	0	0	0
SMO	3	3	3	2	1	1	2	4	5
IBk	0	0	0	0	0	0	0	0	0
J48	1	1	0	0	0	0	0	1	0
JRip	1	1	1	1	1	0	1	1	2
Part	0	0	0	0	0	0	0	1	0
NB	2	1	1	1	1	1	2	3	4
Total	7	6	5	4	3	2	5	10	11

7. Conclusion

In this paper, we have introduced FRNN, a new nearest neighbour classification and prediction approach that exploits the concepts of lower and

upper approximation from fuzzy rough set theory. While it shares the algorithmic simplicity with other NN approaches (IBk, FNN, FRNN-O), we have shown experimentally that our method outperforms them by a comfortable margin, and that it is able to compete with more involved methods including Support Vector Machines.

We have also shown that by replacing the traditional lower and upper approximation by their VQRS counterparts to obtain VQNN, additional resilience can be achieved in the presence of noisy data. Our experiments demonstrate that under normal (non-noisy) conditions, VQNN performs statistically equivalent to FRNN; when noise is added, VQNN soon starts to outperform FRNN, obtaining peek performance when around 25% of the decision values are corrupted with noise. This is a very promising result, and the first clear-cut proof for the noise-tolerant capacities attributed to the VQRS model in [10].

For our future work, we plan to investigate more involved ways of utilizing the information contained in the lower and upper approximations, and of optimizing the fuzzy quantifiers in the VQRS definitions in function of the dataset at hand. We will also look into the integration of our classification/prediction approach with fuzzy-rough feature selection methods, such as [9].

One limitation of the approach is that there is currently no way of dealing with data possessing missing values. An initial attempt at tackling this problem for the task of fuzzy-rough feature selection is given in [23] where an interval-valued approach is adopted. A similar approach could be employed here by using an interval-valued similarity relation and extending both FRNN and VQNN via interval-valued fuzzy-rough sets.

Acknowledgment

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

References

[1] D. Aha, "Instance-based learning algorithm", *Machine Learning*, vol. 6, pp. 37–66, 1991.

[2] V. Suresh Babu, P. Viswanath, "Rough-fuzzy weighted K-nearest leader classifier for large data sets," *Pattern Recognition*, vol. 42, no. 9, pp. 1719–1731, 2009

[3] A. Bargiela, W. Pedrycz, *Granular Computing. An introduction.* Kluwer Academic Publishers, 2002.

[4] R.B. Bhatt and M.Gopal, "FRID: Fuzzy-Rough Interactive Dichotomizers," *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'04)*, pp. 1337–1342, 2004.

[5] H. Bian and L. Mazlack, "Fuzzy-Rough Nearest-Neighbor Classification Approach," *Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, pp. 500–505, 2003.

[6] C.L. Blake, C.J. Merz, *UCI Repository of Machine Learning Databases.* Irvine, University of California, 1998. http://www.ics.uci.edu/˜mlearn/

[7] W.W. Cohen, "Fast Effective Rule Induction," *Proc. 12th Int. Conf. on Machine Learning*, 115–123, 1995.

[8] C. Cornelis, R. Jensen, "A Noise-tolerant Approach to Fuzzy-Rough Feature Selection," *Proceedings of the 17th International Conference on Fuzzy Systems (FUZZ-IEEE08)*, pp. 1598–1605, 2008.

[9] C. Cornelis, R. Jensen, G. Hurtado Martín, "Attribute Selection with Fuzzy Decision Reducts," *Information Sciences*, vol. 180(2), 209–224, 2010.

[10] C. Cornelis, M. De Cock and A. Radzikowska, "Vaguely Quantified Rough Sets,", *Proc. 11th Int. Conf. on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC2007), Lecture Notes in Artificial Intelligence 4482*, 87–94, 2007.

[11] M. De Cock, E.E. Kerre, "On (Un)suitable Fuzzy Relations to Model Approximate Equality", *Fuzzy Sets and Systems*, vol. 133(2), 137–153, 2003.

[12] D. Dubois, H. Prade, "Rough fuzzy sets and fuzzy rough sets," *International Journal of General Systems*, vol. 17, 91–209, 1990.

[13] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[14] A.L. Edwards, *An Introduction to Linear Regression and Correlation*, San Francisco, CA: W. H. Freeman, 1976.

[15] 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.

[16] S. Greco, M. Inuiguchi, and R. Slowinski, "Fuzzy rough sets and multiple-premise gradual decision rules," *International Journal of Approximate Reasoning*, vol. 41, pp. 179–211, 2005.

[17] J.W. Grzymala-Busse, J. Stefanowski, "Three discretization methods for rule induction", *International Journal of Intelligent Systems*, vol. 16(1), 29-38 (2001).

[18] T.P. Hong, Y.L. Liou, and S.L. Wang, "Fuzzy rough sets with hierarchical quantitative attributes," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6790–6799, 2009.

[19] N.-C. Hsieh, "Rule Extraction with Rough-Fuzzy Hybridization Method," *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, vol. 5012, pp. 890–895, 2008.

[20] R. Jensen, C. Cornelis, "A New Approach to Fuzzy-Rough Nearest Neighbour Classification," *Proceedings of the 6th International Conference on Rough Sets and Current Trends in Computing*, pp. 310–319, 2008.

[21] R. Jensen, Q. Shen, *Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches*, Wiley-IEEE Press, 2008.

[22] R. Jensen, Q. Shen, "New approaches to fuzzy-rough feature selection," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 824–838, 2009.

[23] R. Jensen, Q. Shen, "Interval-valued Fuzzy-Rough Feature Selection in Datasets with Missing Values", *Proceedings of the 18th International Conference on Fuzzy Systems (FUZZ-IEEE'09)*, pp. 610-615, 2009.

[24] J.M. Keller, M.R. Gray and J.A. Givens, "A fuzzy K-nearest neighbor algorithm," *IEEE Trans. Systems Man Cybernet.*, vol. 15, no. 4, pp. 580-585, 1985.

[25] P. Langley, "Selection of Relevant Features in Machine Learning", *Proc. AAAI Fall Symp. on Relevance*, 1–5, 1994.

[26] S. Liang-yan and C. Li, "A Fast and Scalable Fuzzy-rough Nearest Neighbor Algorithm," *WRI Global Congress on Intelligent Systems*, vol. 4, pp. 311–314, 2009.

[27] H.S. Nguyen, "Discretization Problem for Rough Sets Methods", *1st Int. Conf. on Rough Sets and Current Trends in Computing (RSCTC'98)*, 545–552, 1198.

[28] Z. Pawlak, "Rough sets," *International Journal of Computer and Information Sciences*, vol. 11(5), 341–356, 1982.

[29] Z. Pawlak, *Rough Sets — Theoretical aspects of reasoning about data.* Kluwer Academic Publishers, Dordrecht, Netherlands, 1991.

[30] W. Pedrycz, "Shadowed Sets: Bridging Fuzzy and Rough Sets," In: *Rough Fuzzy Hybridization a New Trend in Decision-making*, S.K. Pal, A. Skowron (eds.), Springer-Verlag, Singapore, pp. 179–199, 1999.

[31] J.R. Quinlan, *C4.5: Programs for Machine Learning*, The Morgan Kaufmann Series in Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[32] A.M. Radzikowska, E.E. Kerre, E.E., "A comparative study of fuzzy rough sets," *Fuzzy Sets and Systems*, vol. 126, 137–156, 2002.

[33] M. Sarkar, "Fuzzy-Rough nearest neighbors algorithm," *Fuzzy Sets and Systems*, vol. 158, pp. 2123–2152, 2007.

[34] Q. Shen and A. Chouchoulas, "A rough-fuzzy approach for generating classification rules," *Pattern Recognition*, vol. 35, no. 11, pp. 2425–2438, 2002.

[35] A.J. Smola and B. Schölkopf, "A Tutorial on Support Vector Regression," NeuroCOLT2 Technical Report Series - NC2-TR-1998-030, 1998.

[36] T. Takagi and M. Sugeno, "Fuzzy identication of systems and its applications tomodeling and control," *IEEE transactions on systems, man, and cybernetics*, vol. 15,no.1, pp. 116–132, 1985.

[37] X. Wang, J. Yang, X. Teng and N. Peng, "Fuzzy-Rough Set Based Nearest Neighbor Clustering Classification Algorithm," *Lecture Notes in Computer Science*, vol. 3613/2005, pp. 370–373, 2005.

[38] X. Wang, E.C.C. Tsang, S. Zhao, D. Chen and D.S. Yeung, "Learning fuzzy rules from fuzzy samples based on rough set technique," *Information Sciences*, vol. 177, no. 20, pp. 4493–4514, 2007.

[39] Y Wang, A new approach to fitting linear models in high dimensional spaces, PhD Thesis, Department of Computer Science, University of Waikato. 2000.

[40] I.H. Witten and E. Frank, "Generating Accurate Rule Sets Without Global Optimization," *Proceedings of the 15th International Conference on Machine Learning*, Morgan Kaufmann Publishers, San Francisco, 1998.

[41] I.H. Witten, E. Frank, *Data Mining: Practical Machine Learning Tools with Java Implementations*. Morgan Kaufmann, San Francisco, 2000.

[42] L.A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, 338–353, 1965.

[43] L.A. Zadeh, "A Computational Approach to Fuzzy Quantifiers in Natural Languages," *Computers and Mathematics with Applications*, Vol. 9, 149–184, 1983.

[44] L.A. Zadeh, "Soft Computing and Fuzzy Logic," *IEEE Software*, vol. 11(6), 48–56, 1994.

[45] W. Ziarko, "Variable precision rough set model", *Journal of Computer and System Sciences*, vol. 46, 39-59, 1993.

[46] W. Ziarko, "Decision Making with Probabilistic Decision Tables", *Proc. 7th Int. Workshop on New Directions in Rough Sets, Data Mining, and Granular-Soft Computing (RSFDGrC'99)* , 463-471, 1999.

[47] W. Ziarko, "Set approximation quality measures in the variable precision rough set model," *Soft Computing Systems: Design, Management and Applications (A. Abraham, J. Ruiz-del-Solar, M. Koppen, eds.)*, IOS Press, 442–452, 2002.

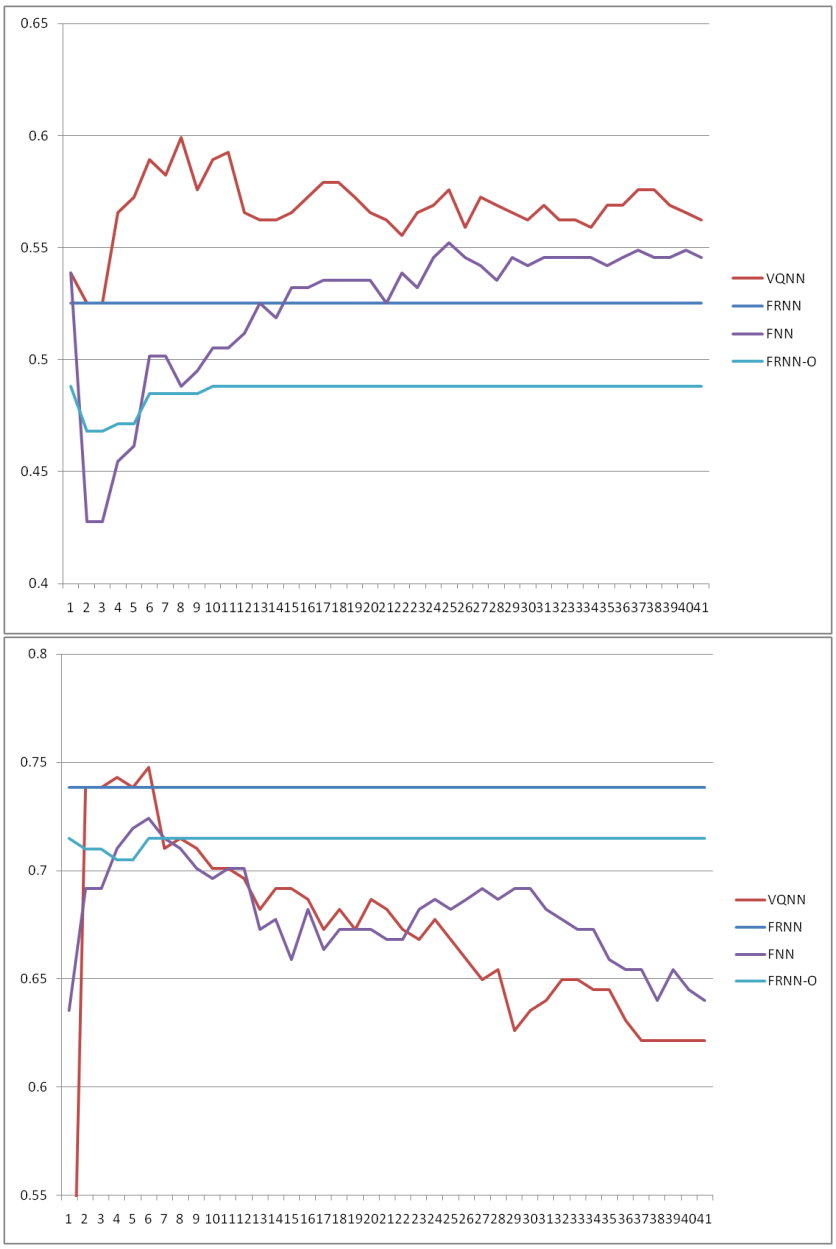

Figure 1: K nearest neighbours vs classification accuracy: `Cleveland` and `Glass` data

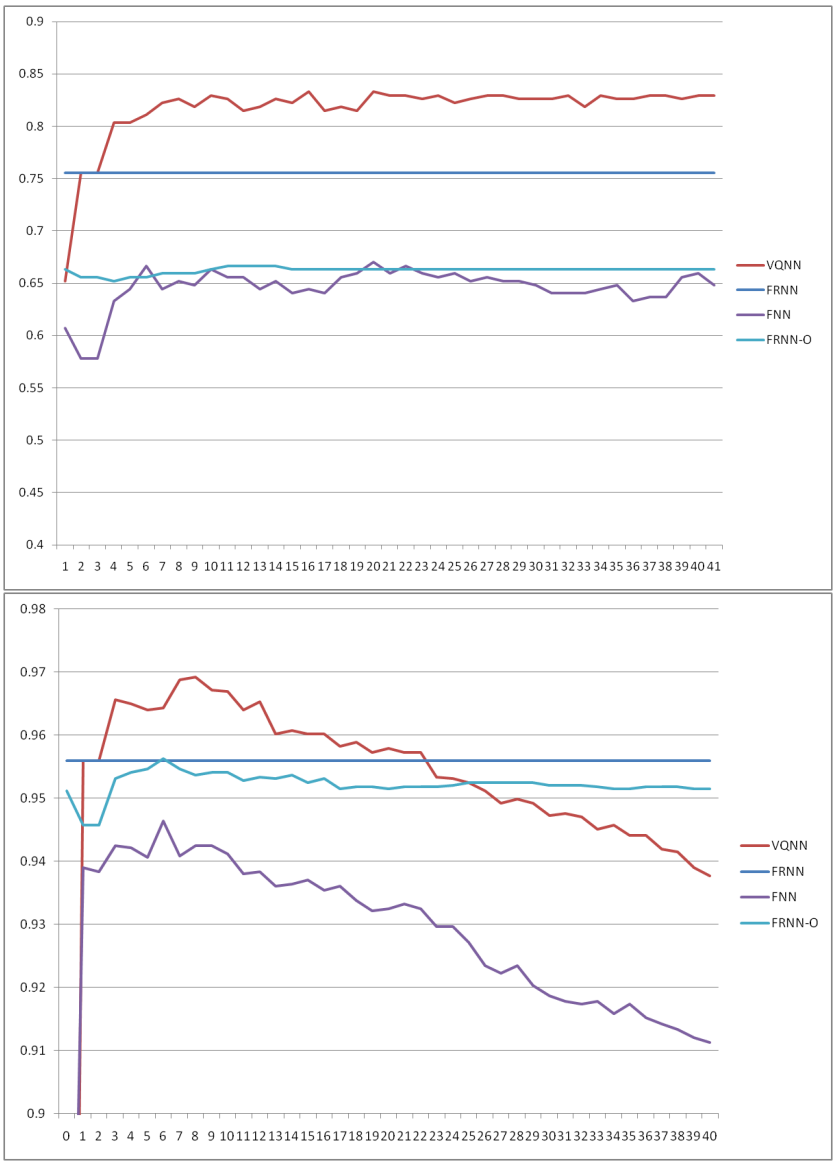

Figure 2: K nearest neighbours vs classification accuracy: `Heart` and `Letter` data

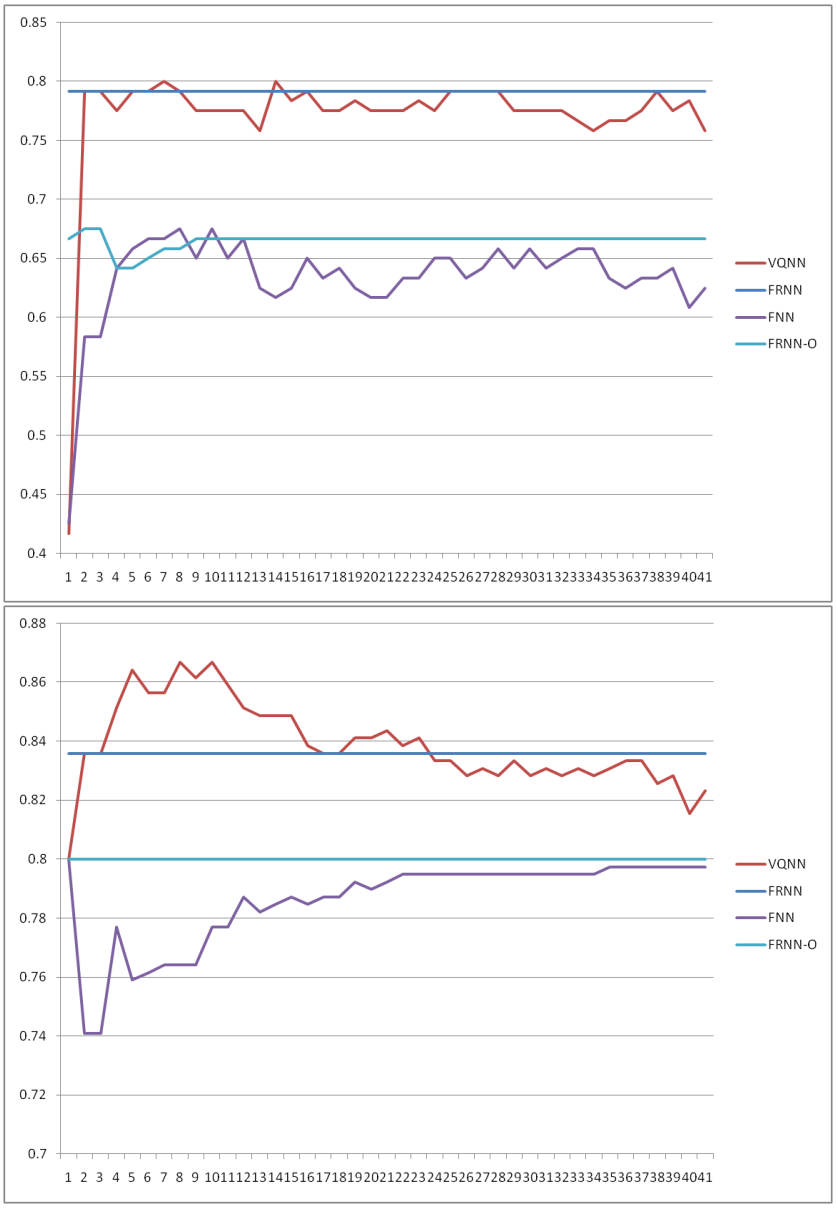

Figure 3: K nearest neighbours vs classification accuracy: `Olitos` and `Water 2` data

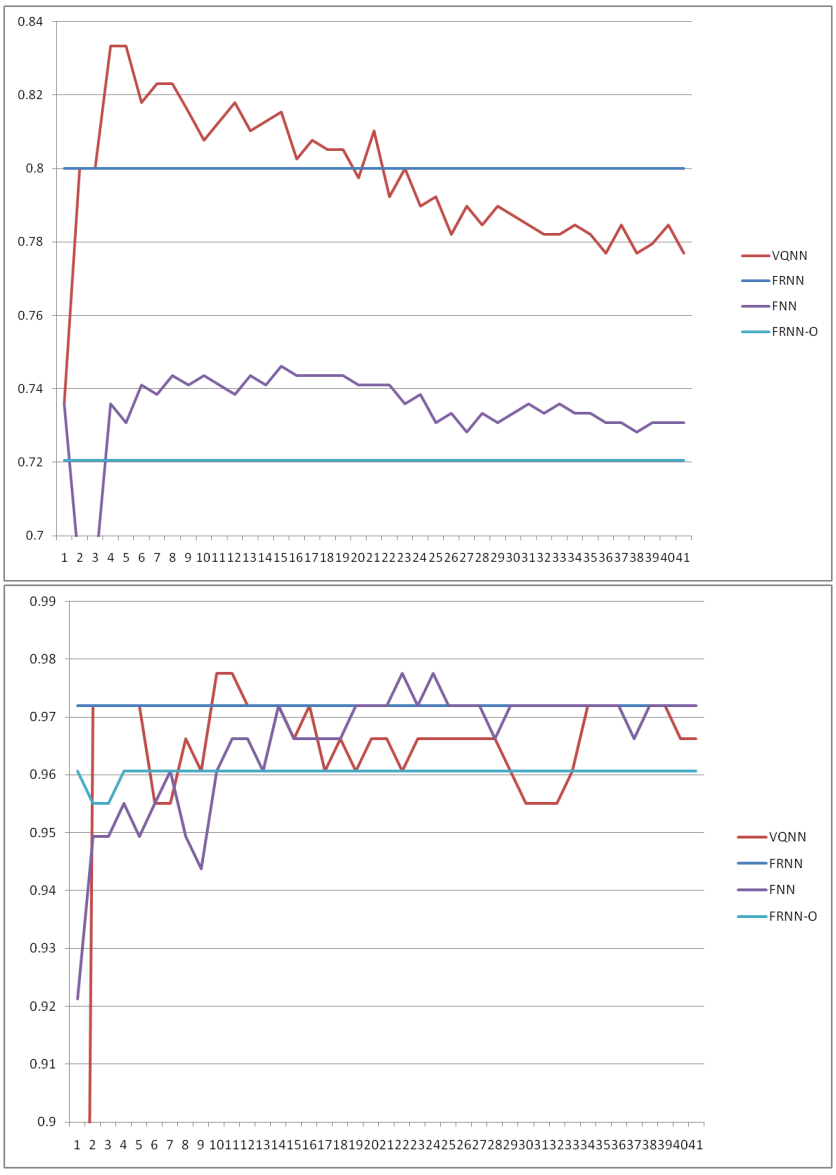

Figure 4: K nearest neighbours vs classification accuracy: `Water 3` and `Wine` data