PRIFYSGOL
ABERYSTWYTH
UNIVERSITY

**Aberystwyth University**

*Multi-functional nearest-neighbour classification*

Qu, Yanpeng; Shang, Changjing; MacParthaláin, Neil; Wu, Wei; Shen, Qiang

CrossMark

# Multi-functional nearest-neighbour classification

Yanpeng Qu[1] · Changjing Shang[2] · Neil Mac Parthaláin[2] · Wei Wu[3] · Qiang Shen[2]

**Abstract** The $k$ nearest-neighbour ($k$NN) algorithm has enjoyed much attention since its inception as an intuitive and effective classification method. Many further developments of $k$NN have been reported such as those integrated with fuzzy sets, rough sets, and evolutionary computation. In particular, the fuzzy and rough modifications of $k$NN have shown significant enhancement in performance. This paper presents another significant improvement, leading to a multi-functional nearest-neighbour (MFNN) approach which is conceptually simple to understand. It employs an aggregation of fuzzy similarity relations and class memberships in playing the critical role of decision qualifier to perform the task of classification. The new method offers important adaptivity in dealing with different classification problems by nearest-neighbour classifiers, due to the large and variable choice of available aggregation methods and similarity metrics. This flexibility allows the proposed approach to be implemented in a variety of forms. Both theoretical analysis and empirical evaluation demonstrate that conventional $k$NN and fuzzy nearest neighbour, as well as two recently developed fuzzy-rough nearest-neighbour algorithms can be considered as special cases of MFNN. Experimental results also confirm that the proposed approach works effectively and generally outperforms many state-of-the-art techniques.

**Keywords** Aggregation · Classification · Nearest-neighbour · Similarity relation

✉ Yanpeng Qu
  yanpengqu@dlmu.edu.cn

  Changjing Shang
  cns@aber.ac.uk

  Neil Mac Parthaláin
  ncm@aber.ac.uk

  Wei Wu
  wuweiw@dlut.edu.cn

  Qiang Shen
  qqs@aber.ac.uk

[1] Information Science and Technology College, Dalian Maritime University, Dalian 116026, China

[2] Department of Computer Science, Institute of Mathematics, Physics and Computer Science, Aberystwyth University, Aberystwyth, Ceredigion SY23 3DB, UK

[3] School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China

## 1 Introduction

Classification systems have played an important role in many application problems, including design, analysis, diagnosis and tutoring (Duda et al. 2001). The goal of developing such a system is to find a model that minimises classification error on data that have not been used during the learning process. Generally, a classification problem can be solved from a variety of perspectives, such as probability theory (Kolmogorov 1950) [e.g. Bayesian networks (John and Langley 1995)], decision tree learning (Breiman et al. 1984) [e.g. C4.5 (Quinlan 1993)] and instance-based learning [e.g. $k$ nearest neighbours or $k$NN (Cover and Hart 1967)]. In particular, variations of $k$NN have been successfully applied to a wide range of real-world problems. It is generally recognised that such an instance-based learning is both practically more effective and intuitively more realistic than many other learning classifier schemes (Daelemans and den Bosch 2005).

Central to the $k$NN approach and its variations is a nonlinear classification technique for categorising objects based on the $k$ closest training objects of a given feature space (Cover and Hart 1967). As a type of instance-based learning, it works by assigning a test object to the decision class that is most common amongst its $k$ nearest neighbours, i.e. the $k$ training objects that are *closest* to the test object. A fuzzy extension of $k$NN is proposed in Keller et al. (1985), known as fuzzy nearest neighbours (FNN), which exploits the varying membership degrees of classes embedded in the training data objects, in order to improve classification performance. Also, an ownership function (Sarkar 2007) has been integrated with the FNN algorithm, producing class confidence values that do not necessarily sum up to one. This method uses fuzzy-rough sets (Dubois and Prade 1992; Yao 1998) (and is abbreviated to FRNN-O hereafter), but it does not utilise the central concepts of lower and upper approximations in rough set theory (Pawlak 1991).

Fuzzy-rough nearest neighbour (FRNN) (Jensen and Cornelis 2011) further extends the $k$NN and FNN algorithms, by using a single test object's nearest neighbours to construct the fuzzy upper and lower approximations for each decision class. The approach offers many different ways in which to construct the fuzzy upper and lower approximations. These include the traditional implicator or $T$-norm-based models (Radzikowska and Kerre 2002), as well as more advanced methods that utilise vaguely quantified rough sets (VQRS) (Cornelis et al. 2007). Experimental results show that a nearest-neighbour classifier based on VQRS, termed VQNN, performs robustly in presence of noisy data. However, the mathematical sophistication required to understand the concepts underlying this and related techniques often hinders their applications.

This paper presents a multi-functional nearest-neighbour (MFNN) classification approach, in order to strengthen the efficacy of the existing advanced nearest-neighbour techniques. An aggregation mechanism for both fuzzy similarity and class membership measured over selected nearest neighbours is employed to act as the decision qualifier. Many similarity metrics [e.g. fuzzy tolerance relations (Das et al. 1998), fuzzy $T$-equivalence relations (Baets and Mesiar 1998, 2002), Euclidean distance, etc] and aggregation operators [e.g. $S$-norms, cardinality measure, *Addition* operator and OWA (Yager 1988), etc] may be adopted for such application. Furthermore, this paper provides a theoretical analysis, showing that with specific implementation of: the aggregator, the similarity relation and the class membership function, FRNN, VQNN, $k$NN and FNN all become particular instances of the MFNN. This observation indicates that the MFNN algorithm grants a flexible framework to the existing nearest-neighbour classification methods. That is, this work helps to ensure that the resulting MFNN is of good adaptivity in dealing with different classification problems given

such a wide range of potential choices. The performance of the proposed novel approach is evaluated through a series of systematic experimental investigations. In comparison with alternative nearest-neighbour methods, including $k$NN, FNN, FRNN-O, and other state-of-the-art classifiers such as: PART (Witten and Frank 1998, 2000), and J48 (Quinlan 1993), versions of MFNN that are implemented with commonly adopted similarity metrics and aggregation operators generally offer improved classification performance.

The remainder of this paper is organised as follows. The theoretical foundation for the multi-functional nearest-neighbour approach is introduced in Sect. 2, including their properties of MFNN and a worked example. The relationship between MFNN and the FRNN and VQNN algorithms is analysed in Sect. 3, supported with theoretical proofs, and that between MFNN and the $k$NN and FNN algorithms is discussed in Sect. 4. The proposed approach is systematically compared with the existing work through experimental evaluations in Sect. 5. Finally, the paper is concluded in Sect. 6, together with a discussion of potential further research.

## 2 Multi-functional nearest-neighbour classifier

For completeness, the $k$NN (Cover and Hart 1967) and fuzzy nearest-neighbour (FNN) (Keller et al. 1985) techniques are briefly recalled first. The multi-functional nearest-neighbour classification is then presented, including its properties and a worked example.

Notationally, in the following, $\mathbb{U}$ denotes a given training dataset that involves a set of features $P$; $y$ denotes the object to be classified; and $a(x)$ denotes the value of an attribute $a, a \in P$, for an instance $x \in \mathbb{U}$.

### 2.1 *K* nearest neighbour

The $k$ nearest-neighbour ($k$NN) classifier (Cover and Hart 1967) works by assigning a test object to the decision class that is most common amongst its $k$ nearest neighbours. Usually, the following Euclidean distance is used to measure the closeness of two data instances:

$$d(x, y) = \sqrt{\sum_{a \in P} (a(x) - a(y))^2}. \tag{1}$$

$k$NN is a simple classification approach without any presumptions on the data. Due to its local nature it has low bias. More specifically, if $k = 1$, the error rate of $k$NN asymptotically never exceeds twice the optimal Bayes error rate (Duda et al. 2001).

## 2.2 Fuzzy nearest neighbour

With $k$NN, each of the selected neighbours is assumed to be equally important, when assigning a class to the target instance. Clearly, this assumption does not hold for many problems. To address this defect, in Keller et al. (1985), a fuzzy extension of $k$NN is proposed, known as fuzzy nearest neighbours (FNN). It considers fuzzy memberships of classes, allowing more flexibility in determining the class assignments through exploiting the fuzziness embedded in the training data. In FNN, after choosing the $k$ nearest neighbours of the unclassified object $y$, the classification result is decided by the maximal decision qualifier:

$$\frac{\sum_{x \in N} \left( d(x, y)^{-\frac{2}{m-1}} \right) \mu_X(x)}{\sum_{x \in N} \left( d(x, y)^{-\frac{2}{m-1}} \right)}, \tag{2}$$

where $N$ represents the set of $k$ nearest neighbours, $d(x, y)$ is the Euclidean distance as of (1), $\mu_X(x)$ denotes the membership function of class $X$, and the parameter $m$ determines how heavily the distance measure is weighted in computing the contribution of each neighbour to the corresponding membership value. As with typical applications of FNN, in this work $m = 2$ unless otherwise stated.

## 2.3 Multi-functional nearest-neighbour algorithm

The multi-functional nearest neighbour (MFNN) method is herein proposed in order to further develop the existing nearest-neighbour classification algorithms. At the highest level, as with the original $k$NN, this approach is conceptually rather simple. It involves a process of three key steps:

1. The fuzzy similarity between the test object and any existing (training) data is calculated, and the $k$ nearest neighbours are selected according to the $k$ greatest resulting similarities. This step is popularly applied for most nearest-neighbour-based methods.
2. The aggregation of fuzzy similarities and the class membership degrees obtained from the $k$ nearest neighbours is computed using the decision qualifier, subject to the choice of a certain aggregation operator. The resulting aggregation generalises the indicator in making decision by a certain nearest-neighbour-based method, such as $k$NN or FNN, etc. This may help to increase the quality of the classification results (see later).
3. Based on the drive to achieve the maximum correctness in decision-making, the final classification decision is drawn on the basis of the aggregation as the one that returns the highest value for the decision qualifier.

The pseudocode for MFNN is presented in Algorithm 1. In this pseudocode, the symbol $A$ stands for the aggregation method applied to similarity measures; $\mu_{R_P}(x, y)$ is a similarity relation induced by the subset of features $P$; $\mu_X(y)$ is the membership function of class $X$, which may be fuzzy or crisp (i.e., $\mu_X(y)$ is 1 if the class of $y$ is $X$ and 0 otherwise) in general. Indeed, for most classification problems, what is interested is to determine which class a given object definitely belongs to and hence the crisp-valued class membership is frequently used.

---

**ALGORITHM 1:** Multi-functional nearest-neighbour algorithm

**Input**: ;
$\mathbb{U}$, the training set;
$\mathcal{C}$, the set of decision classes;
$y$, the object to be classified.
**Output**: $Class$, the classification result for $y$.
$N \leftarrow$ get nearest neighbours $(y, k)$;
$\tau \leftarrow 0, Class \leftarrow \emptyset$;
**for** $\forall X \in \mathcal{C}$ **do**
　**if** $\underset{x \in N}{A}(\mu_{R_P}(x, y)\mu_X(x)) \geq \tau$ **then**
　　$Class \leftarrow X$;
　　$\tau \leftarrow \underset{x \in N}{A}(\mu_{R_P}(x, y)\mu_X(x))$.
　**end**
**end**

---

## 2.4 Properties of MFNN

There are potentially a good number of choices for measuring similarity, such as fuzzy tolerance relations (Das et al. 1998), fuzzy $T$-equivalence relations (Baets and Mesiar 1998, 2002), Euclidean distance, cosine function. Also, there exist various means to perform aggregation of the similarities, such as $S$-norms, cardinality measure, and weighted summation, including *Addition* operator and OWA (Yager 1988). In particular, the following lists those commonly used $S$-norms:

– *Łukasiewicz $S$-norm*: $S(x, y) = \min(x + y, 1)$;
– *Gödel $S$-norm*: $S(x, y) = \max(x, y)$;
– *Algebraic $S$-norm*: $S(x, y) = (x + y) - (x * y)$;
– *Einstein $S$-norm*: $S(x, y) = (x + y)/(1 + x * y)$.

Thus, the MFNN algorithm can be of high flexibility in implementation.

As stated earlier, the combination or aggregation of the measured similarities between objects and class memberships plays the role of decision qualifier in MFNN. However, employing different aggregators may have different impacts upon the ability of suppressing noisy data. For instance, empirically (see later), using the *Addition* operator, MFNN can achieve more robust performance in the presence of

noise, than those implementations which utilise *S*-norms. This may be due to the fact that the *Addition* operator deals with information embedded in the data in an accumulative manner. However, the combination of similarities via *S*-norms is constrained to the interval [0, 1]. Thus, the weight of the contribution of the noise-free data has less overall influence upon the classification outcome. This observation indicates that the framework of MFNN allows adaptation to suit different classification problems given such a wide choice of the aggregators and similarity metrics.

Note that for the MFNN algorithm with crisp class membership, of all the typical *S*-norms, the *Gödel S*-norm is the most notable in terms of its behaviour. This is because of the means by which the *k* nearest neighbours are generated. That is, the maximum similarity between the test object and the existing data in the universe is also the maximum amongst the *k* nearest neighbours. Therefore, with the *Gödel S*-norm, MFNN always classifies a test object into the class where a sample (i.e., the class membership degree is one) has the highest similarity to the test object regardless of the number of nearest neighbours. In other words, when using the *Gödel S*-norm and crisp class membership, the classification accuracy of MFNN is not affected by the choice of the value for *k*.

Computationally, there are two loops in the MFNN algorithm: one to iterate through the classes and another to iterate through the neighbours. Thus, in the worst case (where the nearest neighbourhood covers the entire universe of discourse $\mathbb{U}$), the complexity of MFNN is $O(|\mathcal{C}| \cdot |\mathbb{U}|)$.

## 2.5 Worked example

A simple worked example is presented here in order to illustrate the MFNN algorithm. As summarised in Table 1, it employs a dataset with 3 real-valued conditional attributes (*a*, *b* and *c*) and a single discrete-valued crisp decision attribute (*q*), as the training data. The two objects contained within Table 2 are employed as the test data for classification, which have the same numbers of conditional and decision attributes.

Following the procedure of the MFNN algorithm, the first step is to calculate the similarities for all decision classes ($\forall x \in D_q$ where $D_q$ stands for the domain of the decision

**Table 1** Training data for the example

| Object | a | b | c | q |
|---|---|---|---|---|
| 1 | −0.4 | 0.2 | −0.5 | Yes |
| 2 | −0.4 | 0.1 | −0.1 | No |
| 3 | 0.2 | −0.3 | 0 | No |
| 4 | 0.2 | 0 | 0 | Yes |

**Table 2** Test data for the example

| Object | a | b | c | q |
|---|---|---|---|---|
| t1 | 0.3 | −0.3 | 0 | No |
| t2 | −0.3 | 0.3 | −0.3 | Yes |

attribute, i.e., {*yes,* no}). For simplicity, the class membership is herein set to be crisp-valued. There are four objects that each belong to one of the two classes. Using the *Einstein S*-norm, *Łukasiewicz T*-norm (max($x + y − 1$, 0)) (Borkowski 1970) and the following similarity metric [as defined in Radzikowska and Kerre (2002)]:

$$\mu_{R_P}(x, y) = T_{a \in P}\{\mu_{R_a}(x, y)\}, \tag{3}$$

the similarity of each test object can then be compared to all of the data objects in the training set. In (3), $\mu_{R_a}(x, y)$ is the degree to which objects *x* and *y* are similar with regard to feature *a* and may be defined as (Jensen and Shen 2009):

$$\mu_{R_a}(x, y) = 1 - \frac{|a(x) - a(y)|}{|a_{\max} - a_{\min}|}. \tag{4}$$

Note that for this simple example *all* neighbours are taken into consideration, i.e., $k = |\mathbb{U}|$.

For instance, consider the testing object *t*1:

$$\mu_{R\{P\}}(1, t1) = T(\mu_{R\{a\}}(1, t1), \mu_{R\{b\}}(1, t1), \mu_{R\{c\}}(1, t1))$$
$$= 0$$
$$\mu_{R\{P\}}(2, t1) = T(\mu_{R\{a\}}(2, t1), \mu_{R\{b\}}(2, t1), \mu_{R\{c\}}(2, t1))$$
$$= 0$$
$$\mu_{R\{P\}}(3, t1) = T(\mu_{R\{a\}}(3, t1), \mu_{R\{b\}}(3, t1), \mu_{R\{c\}}(3, t1))$$
$$= 0.86$$
$$\mu_{R\{P\}}(4, t1) = T(\mu_{R\{a\}}(4, t1), \mu_{R\{b\}}(4, t1), \mu_{R\{c\}}(4, t1))$$
$$= 0.26$$

Then, for each class:

$$\mu_{R\{P\}}(\text{yes}, t1) = \frac{\mu_{R\{P\}}(1, t1) + \mu_{R\{P\}}(4, t1)}{1 + \mu_{R\{P\}}(1, t1) * \mu_{R\{P\}}(4, t1)}$$
$$= (0 + 0.26)/(1 + 0 * 0.26) = 0.26 \tag{5}$$
$$\mu_{R\{P\}}(\text{no}, t1) = \frac{\mu_{R\{P\}}(2, t1) + \mu_{R\{P\}}(3, t1)}{1 + \mu_{R\{P\}}(2, t1) * \mu_{R\{P\}}(3, t1)}$$
$$= (0 + 0.86)/(1 + 0 * 0.86) = 0.86 \tag{6}$$

Thus, the combination of similarities for the test object *t*1 with respect to the class label *no* is higher than that to the class *yes*. The algorithm will therefore classify *t*1 as belonging to the class *no*. This procedure can then be repeated for

test object $t2$ in exactly the same way, resulting in $t2$ being classified as belonging to the class *yes*.

## 3 Relationship between MFNN and FRNN/VQNN

The flexible framework of the multi-functional nearest-neighbour algorithm allows it to cover many nearest-neighbour approaches as its special cases. This is demonstrated through the analysis of its relations with the state-of-the-art fuzzy-rough nearest-neighbour classification algorithms below.

### 3.1 Fuzzy-rough nearest-neighbour classification

The original fuzzy-rough nearest-neighbour (FRNN) algorithm was proposed in Jensen and Cornelis (2011). In contrast to approaches such as fuzzy-rough ownership nearest neighbour (FRNN-O) (Sarkar 2007), FRNN employs the central rough set concepts in their fuzzified forms: fuzzy upper and lower approximations. These important concepts are used to determine the assignment of class membership to a given test object.

A fuzzy-rough set (Dubois and Prade 1992; Yao 1998) is defined by two fuzzy sets, obtained by extending the concepts of the upper and lower approximations in crisp rough sets (Pawlak 1991). In particular, the fuzzy upper and lower approximations of a certain object $y$ concerning a fuzzy concept $X$ are defined as follows:

$$\mu_{\underline{R_P}X}(y) = \inf_{x \in \mathbb{U}} I(\mu_{R_P}(x, y), \mu_X(x)) \quad (7)$$

$$\mu_{\overline{R_P}X}(y) = \sup_{x \in \mathbb{U}} T(\mu_{R_P}(x, y), \mu_X(x)) \quad (8)$$

where $I$ is a fuzzy implicator, $T$ is a $T$-norm, $\mu_{R_P}(x, y)$ is the fuzzy similarity defined in (3) and $\mu_X(x)$ is the class membership function. More sophisticated mathematical treatment of these can be found in Jensen and Shen (2009) and Radzikowska and Kerre (2002).

The FRNN algorithm is outlined in Algorithm 2. As a extension of this method, vaguely quantified nearest neighbour (VQNN) has also been developed, as reported in Cornelis et al. (2007). This is achieved with the introduction of the lower and upper approximations of vaguely quantified rough sets (VQRS):

$$\mu_{\underline{R_P}X}^{Q_u}(y) = Q_u \left( \frac{|R_P(x, y) \cap X|}{|R_P(x, y)|} \right)$$

$$= Q_u \left( \frac{\sum_{x \in \mathbb{U}} T(\mu_{R_P}(x, y), \mu_X(x))}{\sum_{x \in \mathbb{U}} \mu_{R_P}(x, y)} \right), \quad (9)$$

$$\mu_{\overline{R_P}X}^{Q_l}(y) = Q_l \left( \frac{|R_P(x, y) \cap X|}{|R_P(x, y)|} \right)$$

$$= Q_l \left( \frac{\sum_{x \in \mathbb{U}} T(\mu_{R_P}(x, y), \mu_X(x))}{\sum_{x \in \mathbb{U}} \mu_{R_P}(x, y)} \right). \quad (10)$$

The pair $(Q_u, Q_l)$ are fuzzy quantifiers (Cornelis et al. 2007), with each element being an increasing $[0, 1] \to [0, 1]$ mapping.

---

**ALGORITHM 2:** Fuzzy-rough nearest-neighbour algorithm

**Input**: ;
$\mathbb{U}$, the training set;
$\mathcal{C}$, the set of decision classes;
$y$, the object to be classified.
**Output**: *Class*, the classification result for $y$.
$N \leftarrow$ get nearest neighbour$(y, k)$;
$\tau \leftarrow 0, Class \leftarrow \emptyset$;
**for** $\forall X \in \mathcal{C}$ **do**
    **if** $(\mu_{\underline{R_P}X}(y) + \mu_{\overline{R_P}X}(y))/2 \geq \tau$ **then**
        $Class \leftarrow X$;
        $\tau \leftarrow (\mu_{\underline{R_P}X}(y) + \mu_{\overline{R_P}X}(y))/2$.
    **end**
**end**

---

### 3.2 FRNN and VQNN as special instances of MFNN

Conceptually, MFNN is much simpler without the need of directly involving complicated mathematical definitions such as the above. However, its generality covers both FRNN and VQNN as its specific cases. Of the many methods for aggregating similarities, *Gödel* (*Maximum*) *S*-norm and the *Addition* operator are arguably amongst those which are most commonly used. By using these two aggregators, two particular implementations of MFNN can be devised, denoting them as MFNN_G (MFNN with *Gödel S*-norm) and MFNN_A (MFNN with *Addition* operator). If the fuzzy similarity defined in (3) and the crisp class membership function are employed to implement MFNN_G and MFNN_A, then, these two methods are, respectively, equivalent to FRNN and VQNN, in terms of classification outcomes. The proofs are given as follows.

**Theorem 1** *With the use of an identical fuzzy similarity metric, MFNN_G achieves the same classification accuracy as FRNN.*

*Proof* Given a set $\mathbb{U}$ and a test object $y$, consider using FRNN for classification first. Within the selected nearest neighbours, let $M$ be the class that $\exists x^* \in M$, such that $\mu_{R_P}(x^*, y) = \max_{x \in \mathbb{U}}\{\mu_{R_P}(x, y)\}$. For any $a \in \mathbb{R}$, $I(a, 1) = 1$ and

$T(a, 0) = 0$. Thus, due to the crisp membership function, for the decision concept $X = M$, the lower and upper approximations are reduced to:

$$\mu_{\underline{R_P}M}(y) = \inf\left(\inf_{x\in M} I(\mu_{R_P}(x, y), \mu_M(x)),\right.$$
$$\left.\inf_{x\in\mathbb{U}-M} I(\mu_{R_P}(x, y), \mu_M(x))\right)$$
$$= \inf_{x\in\mathbb{U}-M} I(\mu_{R_P}(x, y), 0), \tag{11}$$

$$\mu_{\overline{R_P}M}(y) = \sup\left(\sup_{x\in M} T(\mu_{R_P}(x, y), \mu_M(x)),\right.$$
$$\left.\sup_{x\in\mathbb{U}-M} T(\mu_{R_P}(x, y), \mu_M(x))\right)$$
$$= \sup_{x\in M} T(\mu_{R_P}(x, y), 1). \tag{12}$$

According to the properties of implicators and $T$-norms, for any given $y$, $I(x, y)$ is monotonically decreasing for $x$, and $T(x, y)$ is monotonically increasing for $x$. In this case, (11) and (12) can be further simplified to:

$$\mu_{\underline{R_P}M}(y) = I\left(\max_{x\in\mathbb{U}-M}\{\mu_{R_P}(x, y)\}, 0\right), \tag{13}$$

$$\mu_{\overline{R_P}M}(y) = T\left(\max_{x\in M}\{\mu_{R_P}(x, y)\}, 1\right)$$
$$= T(\mu_{R_P}(x^*, y), 1). \tag{14}$$

Similarly, for the decision concept $X = L$, where $L$ is an arbitrary class different from $M$, the following hold:

$$\mu_{\underline{R_P}L}(y) = \inf_{x\in\mathbb{U}-L} I(\mu_{R_P}(x, y), 0), \tag{15}$$

$$\mu_{\overline{R_P}L}(y) = \sup_{x\in L} T(\mu_{R_P}(x, y), 1). \tag{16}$$

In the same way as (11) and (12), (15) and (16) are further reduced to:

$$\mu_{\underline{R_P}L}(y) = I\left(\max_{x\in\mathbb{U}-L}\{\mu_{R_P}(x, y)\}, 0\right)$$
$$= I(\mu_{R_P}(x^*, y), 0), \tag{17}$$

$$\mu_{\overline{R_P}L}(y) = T\left(\max_{x\in L}\{\mu_{R_P}(x, y)\}, 1\right). \tag{18}$$

Since

$$\mu_{R_P}(x^*, y) \geq \max_{x\in\mathbb{U}-M}\{\mu_{R_P}(x, y)\} \tag{19}$$

$$\mu_{R_P}(x^*, y) \geq \max_{x\in L}\{\mu_{R_P}(x, y)\} \tag{20}$$

according to the monotonic properties of implicators and $T$-norms, it can be derived that

$$\mu_{\underline{R_P}M}(y) \geq \mu_{\underline{R_P}L}(y), \tag{21}$$

$$\mu_{\overline{R_P}M}(y) \geq \mu_{\overline{R_P}L}(y). \tag{22}$$

Therefore,

$$\frac{\mu_{\underline{R_P}M}(y) + \mu_{\overline{R_P}M}(y)}{2} \geq \frac{\mu_{\underline{R_P}L}(y) + \mu_{\overline{R_P}L}(y)}{2}. \tag{23}$$

Because of the arbitrariness of class $L$, following the method of FRNN, $y$ will be classified into class $M$.

Now, consider the use of MFNN_G. Given the same assumptions and parameters as assumed above, because MFNN_G is implemented by the same fuzzy similarity metric as FRNN, these two methods will share the identical nearest neighbours as well. And for $x^* \in M$, it can be derived that

$$A_{x\in M}\mu_{R_P}(x, y)\mu_M(x) = A_{x\in M}\mu_{R_P}(x, y) \cdot 1$$
$$= \max_{x\in M}\{\mu_{R_P}(x, y)\}$$
$$= \mu_{R_P}(x^*, y)$$
$$= \max_{x\in\mathbb{U}}\{\mu_{R_P}(x, y)\}$$
$$\geq \max_{x\in L}\{\mu_{R_P}(x, y)\}$$
$$= A_{x\in L}\mu_{R_P}(x, y)\mu_L(x) \tag{24}$$

Because of the arbitrariness of class $L$, following the method of MFNN, $y$ will be classified into class $M$. Therefore, the classification outcome of MFNN_G is identical to that of FRNN. □

**Theorem 2** *With the use of an identical fuzzy similarity metric, MFNN_A achieves the same classification accuracy as VQNN.*

*Proof* Given a set $\mathbb{U}$ and a test object $y$, Within the selected nearest neighbours, let $M$ be the class that $\sum_{x\in M}\mu_{R_P}(x, y) = \max_{X\in\mathcal{C}}\{\sum_{x\in X}\mu_{R_P}(x, y)\}$, where $\mathcal{C}$ denotes the set of decision classes.

According to the definition of VQNN, for the decision concept $X = M$, the lower and upper approximations are

$$\mu_{\underline{R_P}M}^{Q_u}(y) = Q_u\left(\frac{\sum_{x\in\mathbb{U}} T(\mu_{R_P}(x, y), \mu_M(x))}{\sum_{x\in\mathbb{U}} \mu_{R_P}(x, y)}\right)$$
$$= Q_u\left(\frac{\sum_{x\in M} T(\mu_{R_P}(x, y), \mu_M(x))}{\sum_{x\in\mathbb{U}} \mu_{R_P}(x, y)}\right.$$
$$\left.+ \frac{\sum_{x\notin M} T(\mu_{R_P}(x, y), \mu_M(x))}{\sum_{x\in\mathbb{U}} \mu_{R_P}(x, y)}\right) \tag{25}$$

$$\mu\frac{Q_l}{R_P M}(y) = Q_l\left(\frac{\sum\limits_{x\in\mathbb{U}}T(\mu_{R_P}(x,y),\mu_M(x))}{\sum\limits_{x\in\mathbb{U}}\mu_{R_P}(x,y)}\right)$$

$$= Q_l\left(\frac{\sum\limits_{x\in M}T(\mu_{R_P}(x,y),\mu_M(x))}{\sum\limits_{x\in\mathbb{U}}\mu_{R_P}(x,y)}\right.$$

$$\left.+\frac{\sum\limits_{x\notin M}T(\mu_{R_P}(x,y),\mu_M(x))}{\sum\limits_{x\in\mathbb{U}}\mu_{R_P}(x,y)}\right) \tag{26}$$

Because $T(a,0)=0$, $T(a,1)=a$ and the class membership is crisp-valued, (25) and (26) can be simplified as

$$\mu\frac{Q_u}{R_P M}(y) = Q_u\left(\frac{\sum\limits_{x\in M}\mu_{R_P}(x,y)}{\sum\limits_{x\in\mathbb{U}}\mu_{R_P}(x,y)}\right) \tag{27}$$

$$\mu\frac{Q_l}{R_P M}(y) = Q_l\left(\frac{\sum\limits_{x\in M}\mu_{R_P}(x,y)}{\sum\limits_{x\in\mathbb{U}}\mu_{R_P}(x,y)}\right). \tag{28}$$

Similarly, for a distinct decision $X = L$, the lower and upper approximation can be, respectively, denoted by

$$\mu\frac{Q_u}{R_P L}(y) = Q_u\left(\frac{\sum\limits_{x\in L}\mu_{R_P}(x,y)}{\sum\limits_{x\in\mathbb{U}}\mu_{R_P}(x,y)}\right) \tag{29}$$

$$\mu\frac{Q_l}{R_P L}(y) = Q_l\left(\frac{\sum\limits_{x\in L}\mu_{R_P}(x,y)}{\sum\limits_{x\in\mathbb{U}}\mu_{R_P}(x,y)}\right). \tag{30}$$

Following the definition of VQNN, the fuzzy qualifier in VQNN is an increasing $[0,1]\to[0,1]$ relation. In this case, because

$$\sum\limits_{x\in M}\mu_{R_P}(x,y) \ge \sum\limits_{x\in L}\mu_{R_P}(x,y), \tag{31}$$

it can be established that

$$\mu\frac{Q_u}{R_P M}(y) \ge \mu\frac{Q_u}{R_P L}(y) \tag{32}$$

$$\mu\frac{Q_l}{R_P M}(y) \ge \mu\frac{Q_l}{R_P L}(y), \tag{33}$$

Then, obviously,

$$\frac{\mu\frac{Q_u}{R_P M}(y) + \mu\frac{Q_l}{R_P M}(y)}{2} \ge \frac{\mu\frac{Q_u}{R_P L}(y) + \mu\frac{Q_l}{R_P L}(y)}{2} \tag{34}$$

This concludes that $y$ will always be classified into Class $M$.

For MFNN_A, from the same assumptions and parameters which are used above, due to the identical nearest neighbours to those of VQNN, it can be derived that

$$\mathop{A}_{x\in M}\mu_{R_P}(x,y)\mu_M(x) = \mathop{A}_{x\in M}\mu_{R_P}(x,y)\times 1$$

$$= \sum\limits_{x\in M}\mu_{R_P}(x,y)$$

$$= \max_{X\in\mathcal{C}}\left\{\sum\limits_{x\in X}\mu_{R_P}(x,y)\right\}$$

$$\ge \sum\limits_{x\in L}\mu_{R_P}(x,y)$$

$$= \mathop{A}_{x\in L}\mu_{R_P}(x,y)\mu_L(x) \tag{35}$$

Because of the arbitrariness of class $L$, according to the definition of MFNN, $y$ will be classified into class $M$. Therefore, the classification outcome of MFNN_A is the same as that of VQNN. $\qquad\square$

It is noteworthy that, implemented by above configuration, MFNN_G and MFNN_A, are equivalent to FRNN and VQNN, respectively, only in the sense that they can achieve the same classification outcome and hence the same classification accuracy. However, the actual underlying predicted probabilities and models of MFNN_G/MFNN_A and FRNN/VQNN are different. Therefore, for other classification performance indicators, such as root-mean-squared error (RMSE), which are determined by the numerical predicted probabilities, the equivalent relationships between MFNN_G /MFNN_A and FRNN/VQNN may not hold.

## 4 Popular nearest-neighbour methods as special cases of MFNN

To further demonstrate the generality of MFNN, this section discusses the relationship between MFNN and $k$ nearest neighbour and that between MFNN and fuzzy nearest neighbour.

### 4.1 *K* nearest neighbour and MFNN

As introduced previously, the $k$ nearest-neighbour ($k$NN) classifier (Cover and Hart 1967) assigns a test object to the class that is best represented amongst its $k$ nearest neighbours.

In implementing MFNN, in addition to the use of the crisp class membership, the cardinality measure can be employed to perform the aggregation. More specifically, the aggregation of the similarities with respect to each class $X$ for the given set $N$ of $k$ nearest neighbours can be defined as:

$$\underset{x \in N}{A} \mu_{R_P}(x, y)\mu_X(x) = \underset{x \in N \cap X}{A} \mu_{R_P}(x, y)$$

$$= \sum_{x \in N \cap X} f(\mu_{R_P}(x, y)), \qquad (36)$$

where the function $f(\cdot) \equiv 1$. In so doing, the aggregation of the similarities and the class membership for each class within the $k$ nearest neighbours is the number of such elements that are nearest to the test object. Therefore, the test object will be classified into the class which is the most common amongst its $k$ nearest neighbours by MFNN. As such, MFNN becomes $k$NN if (the opposite of) the Euclidean distance in (1) is used to measure the similarity as it is used to measure the closeness in $k$NN.

## 4.2 Fuzzy nearest neighbour and MFNN

In fuzzy nearest neighbour (FNN) (Keller et al. 1985), the $k$ nearest neighbours of the test instance are determined first. Then, the test instance is assigned to the class for which the decision qualifier (2) is maximal.

Note that the similarity relation in MFNN can be configured such that

$$\mu_{R_P}(x, y) = d(x, y)^{-\frac{2}{m-1}}, \qquad (37)$$

which is indeed a valid similarity metric. Note also that by setting the aggregator as the weighted summation, with the weights uniformly set to $\frac{1}{\sum_{x \in N} \mu_{R_P}(x,y)}$, the decision indicator of MFNN becomes:

$$\underset{x \in N}{A} \mu_{R_P}(x, y)\mu_X(x)$$

$$= \left( \frac{1}{\sum_{x \in N} \mu_{R_P}(x, y)} \right) \sum_{x \in N} \mu_{R_P}(x, y)\mu_X(x), \qquad (38)$$

where $\mu_X(x)$ is the same class membership used in FNN as well. In so doing, FNN and MFNN will have the identical classification results. For FRNN-O (Sarkar 2007), similar conclusions also hold. The analyses regarding both cases are parallel to the above and therefore, omitted here.

## 5 Experimental evaluation

This section presents a systematic evaluation of MFNN experimentally. The results and discussions are divided into four different parts, after an introduction to the experimental set-up. The first part investigates the influence of the number of nearest neighbours on the MFNN algorithm and how this may affect classification performance. The second

compares MFNN with five other nearest-neighbour methods in term of classification accuracy. It also demonstrates empirically how MFNN may be equivalent to the fuzzy-rough set-based FRNN and VQNN approaches, supporting the earlier formal proofs. The third and fourth parts provide a comparative investigation of the performance of several versions of the MFNN algorithm against four state-of-the-art classifier learners. Once again, these comparisons are made with regard to the classification accuracy.

### 5.1 Experimental set-up

Sixteen benchmark datasets obtained from Armanino et al. (1989) and Blake and Merz (1998) are used for the experimental evaluation. These datasets contain between 120 and 6435 objects with numbers of features ranging from 6 to 649, as summarised in Table 3.

For the present study, MFNN employs the popular relation of (4) and the Kleene-Dienes $T$-norm (Dienes 1949; Kleene 1952) to measure the fuzzy similarity as per (3) in the following experiments. Whilst this does not allow the opportunity to fine-tune individual MFNN methods, it ensures that methods are compared on equal footing. For simplicity, in the following experiments, the class membership functions utilised in MFNN are consistently set to be crisp-valued.

Stratified $10 \times 10$-fold cross-validation (10-FCV) is employed throughout the experimentation. In 10-FCV, an original dataset is partitioned into 10 subsets of data objects. Of these 10 subsets, a single subset is retained as the testing data for the classifier, and the remaining 9 subsets are used for training. The cross-validation process is repeated

**Table 3** Datasets used for evaluation

| Dataset | Objects | Attributes |
| --- | --- | --- |
| Cleveland | 297 | 13 |
| Ecoli | 336 | 7 |
| Glass | 214 | 9 |
| Handwritten | 1593 | 256 |
| Heart | 270 | 13 |
| Liver | 345 | 6 |
| Multifeat | 2000 | 649 |
| Olitos | 120 | 25 |
| Page-block | 5473 | 10 |
| Satellite | 6435 | 36 |
| Sonar | 208 | 60 |
| Water 2 | 390 | 38 |
| Water 3 | 390 | 38 |
| Waveform | 5000 | 40 |
| Wine | 178 | 14 |
| Wisconsin | 683 | 9 |

for 10 times. The 10 sets of results are then averaged to produce a single classifier estimation. The advantage of 10-FCV over random subsampling is that all objects are used for both training and testing, and each object is used for testing only once per fold. The stratification of the data prior to its division into folds ensures that each class label (as far as possible) has equal representation in all folds, thereby helping to alleviate bias/variance problems (Bengio and Grandvalet 2005).

A paired $t$-test with a significance level of 0.05 is employed to provide statistical analysis of the resulting classification accuracy. This is done in order to ensure that results are not discovered by chance. The baseline references for the tests are the classification accuracy obtained by MFNN. The results on statistical significance are summarised in the final line of each table, showing a count of the number of statistically better (v), equivalent ( ) or worse (*) cases for each method in comparison to MFNN. For example, in Table 4, (2/8/6) in the FNN column indicates that MFNN_G performs better than this method on 6 datasets, equivalently to it on 8 datasets, and worse than it in just 2 datasets.

### 5.2 Influence of number of neighbours

As mentioned previously, when using the *Gödel* operator and crisp class membership, the classification accuracy of MFNN is not affected by the selection of the value of $k$. This is not generally the case when other operators are employed. In order to demonstrate this empirically, the impact of different values of $k$ is investigated for MFNN on two datasets, *heart* and *olitos*. The *Einstein* and *Addition* operators as well as the *Gödel* $S$-norm are used to implement MFNN, with the resulting classifiers denoted as MFNN_E, MFNN_A and MFNN_G, respectively. For the investigation described here, $k$ is initially set to $|\mathbb{U}|$ (the total number of the objects in the dataset) and then decremented by $1/30$ of $|\mathbb{U}|$ each time, with an extra round for the case when $k = 1$. This results in 31 runs for each of the two datasets. For each value of $k$, $10 \times 10$-fold cross-validation is performed. The results for these two datasets are shown in Figs. 1 and 2.

It can be seen that MFNN_G is unaffected by the choice of $k$. However, MFNN_E and MFNN_A initially exhibit improvement in classification performance, followed by a degradation for both datasets as the value of $k$ decreases. Therefore, the choice of value for $k$ is an important consideration when using an aggregator other than the *Gödel* operator. Careful off-line selection of an appropriate $k$ is necessary before MFNN is applied (unless MFNN_G is to be used). This conforms to the general findings in the $k$NN literature.

### 5.3 Comparison with other nearest neighbour methods

This section presents a comparison of MFNN with other five nearest-neighbour classification methods: $k$ nearest neigh-
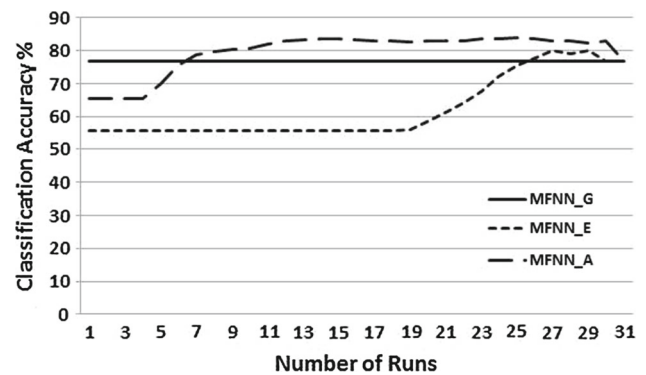


**Fig. 1** Classification accuracy of three MFNNs (*Gödel, Einstein, Addition*) with respect to different $k$ values for the *heart* dataset
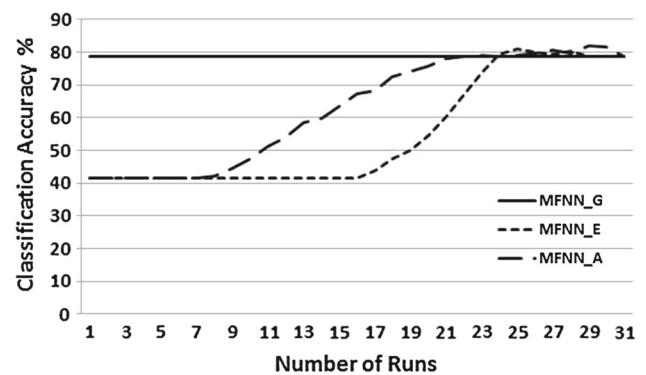


**Fig. 2** Classification accuracy of three MFNNs (*Gödel, Einstein, Addition*) with respect to different $k$ values for the *olitos* dataset

bour ($k$NN) (Cover and Hart 1967), fuzzy nearest neighbour (FNN) (Keller et al. 1985), fuzzy-rough ownership nearest neighbour (FRNN-O) (Sarkar 2007), fuzzy-rough nearest neighbour (FRNN) (Jensen and Cornelis 2011) and vaguely quantified nearest neighbour (VQNN) (Cornelis et al. 2007). In order to provide a fair comparison, all of the results are generated with the value of $k$ set to 10 when implementing MFNN_A, VQNN, FNN, FRNN-O and $k$NN.

Tables 4 and 5 show that as expected, for all datasets, MFNN_G and MFNN_A offer identical classification accuracies to those of FRNN and VQNN, respectively. When compared to FNN, FRNN-O and $k$NN, different implementations of MFNN generally return statistically better or equal results, especially for the *cleveland, heart, olitos, water2,* and *water3* datasets. This is shown in these two tables and also in Table 6 (for MFNN_E). There are very occasional cases where MFNN methods perform less well. In particular, for the *waveform* dataset, the MFNN classifiers fail to perform so well as FNN, but these few results do not reflect the general trend in which MFNN methods outperform the rest.

Comparing the three MFNN classifiers themselves, MFNN_A statistically achieves the best performance, with an exception for the *sonar* dataset. MFNN_E also performs

**Table 4** Classification accuracy: MFNN_G versus others

| Dataset | MFNN_G | FRNN | VQNN | FNN | FRNN-O | kNN |
|---|---|---|---|---|---|---|
| Cleveland | 53.44 | 53.44 | 58.46 | 49.75 | 46.85* | 55.73 |
| Ecoli | 80.57 | 80.57 | 86.85v | 86.55v | 77.95 | 86.20v |
| Glass | 73.54 | 73.54 | 68.95 | 68.57 | 71.70 | 63.23* |
| Handwritten | 91.13 | 91.13 | 91.37 | 91.40 | 89.94* | 90.18 |
| Heart | 76.63 | 76.63 | 82.19v | 66.11* | 66.00* | 81.30 |
| Liver | 62.81 | 62.81 | 66.26 | 69.52 | 62.37 | 61.25 |
| Multifeat | 97.57 | 97.57 | 97.95 | 94.34* | 96.96 | 97.88 |
| Olitos | 78.67 | 78.67 | 80.75 | 63.25* | 67.58* | 81.50 |
| Page-block | 96.04 | 96.04 | 95.99 | 95.94 | 96.53 | 95.19* |
| Satellite | 90.92 | 90.92 | 90.99 | 90.73 | 90.89 | 90.30 |
| Sonar | 85.25 | 85.25 | 79.38* | 73.21* | 85.06 | 75.25 |
| Water2 | 84.38 | 84.38 | 85.15 | 77.97* | 79.79* | 84.26 |
| Water3 | 79.82 | 79.82 | 81.28 | 74.64* | 73.21* | 80.90 |
| Waveform | 73.77 | 73.77 | 81.55v | 83.19v | 79.71v | 80.46v |
| Wine | 97.47 | 97.47 | 97.14 | 96.40 | 95.62 | 96.07 |
| Wisconsin | 96.38 | 96.38 | 96.69 | 97.20 | 96.00 | 96.92 |
| Summary (v//*) | | (0/16/0) | (3/12/1) | (2/8/6) | (1/9/6) | (2/11/3) |

**Table 5** Classification accuracy: MFNN_A versus others

| Dataset | MFNN_A | FRNN | VQNN | FNN | FRNN-O | kNN |
|---|---|---|---|---|---|---|
| Cleveland | 58.46 | 53.44 | 58.46 | 49.75* | 46.85* | 55.73 |
| Ecoli | 86.85 | 80.57* | 86.85 | 86.55 | 77.95 | 86.20 |
| Glass | 68.95 | 73.54 | 68.95 | 68.57 | 71.70 | 62.23* |
| Handwritten | 91.37 | 91.13 | 91.37 | 91.40 | 89.94* | 90.18* |
| Heart | 82.19 | 76.63* | 82.19 | 66.11* | 66.00* | 81.30 |
| Liver | 66.26 | 62.81 | 66.26 | 69.52 | 62.37 | 61.25* |
| Multifeat | 97.95 | 97.57 | 97.95 | 94.34* | 96.96* | 97.88 |
| Olitos | 80.75 | 78.67 | 80.75 | 63.25* | 67.58* | 81.50 |
| Page-block | 95.99 | 96.04 | 95.99 | 95.94 | 96.53 | 95.19* |
| Satellite | 90.99 | 90.92 | 90.99 | 90.73 | 90.89 | 90.30* |
| Sonar | 79.38 | 85.25v | 79.38 | 73.21* | 85.06v | 75.25 |
| Water2 | 85.15 | 84.38 | 85.15 | 77.97* | 79.79* | 84.26 |
| Water3 | 81.28 | 79.82 | 81.28 | 74.64* | 73.21* | 80.90 |
| Waveform | 81.55 | 73.77* | 81.55 | 83.19v | 79.71* | 80.46 |
| Wine | 97.14 | 97.47 | 97.14 | 96.40 | 95.62 | 96.07 |
| Wisconsin | 96.69 | 96.38 | 96.69 | 97.20 | 96.00 | 96.92 |
| Summary (v//*) | | (1/12/3) | (0/16/0) | (1/8/7) | (1/7/8) | (0/11/5) |

slightly better than MFNN_G. This is due to the fact that the classification results gained by MFNN_G only rely on one sample. In this case, MFNN_G is sensitive to noisy data, whilst MFNN_A is significantly more robust in the presence of noisy data.

Overall, considering all the experimental results, this method outperforms all of the existing methods. This makes MFNN_A a potentially good candidate for many classification tasks.

## 5.4 Comparison with the state of the art: use of different aggregators

This section experimentally compares MFNN with several leading classifier learners that represent a cross section of the most popular approaches. For completeness, a brief summary of these methods is provided below.

– PART (Witten and Frank 1998, 2000) generates rules by means of repeatedly creating partial decision trees

**Table 6** Classification accuracy: MFNN_E versus others

| Dataset | MFNN_E | FRNN | VQNN | FNN | FRNN-O | kNN |
|---|---|---|---|---|---|---|
| Cleveland | 53.64 | 53.44 | 58.46 | 49.75 | 46.85* | 55.73 |
| Ecoli | 81.93 | 80.57* | 86.85v | 86.55v | 77.95 | 86.20v |
| Glass | 74.29 | 73.54 | 68.95 | 68.57 | 71.70 | 63.23* |
| Handwritten | 91.20 | 91.13 | 91.37 | 91.40 | 89.94* | 90.18 |
| Heart | 76.70 | 76.63 | 82.19v | 66.11* | 66.00* | 81.30 |
| Liver | 63.07 | 62.81 | 66.26 | 69.52 | 62.37 | 61.25 |
| Multifeat | 97.59 | 97.57 | 97.95 | 94.34* | 96.96 | 97.88 |
| Olitos | 78.83 | 78.67 | 80.75 | 63.25* | 67.58* | 81.50 |
| Page-block | 96.18 | 96.04 | 95.99 | 95.94 | 96.53 | 95.19* |
| Satellite | 91.52 | 90.92* | 90.99 | 90.73* | 90.89* | 90.30* |
| Sonar | 85.35 | 85.25 | 79.38* | 73.21* | 85.06 | 75.25* |
| Water2 | 85.21 | 84.38 | 85.15 | 77.97* | 79.79* | 84.26 |
| Water3 | 80.77 | 79.82 | 81.28 | 74.64* | 73.21* | 80.90 |
| Waveform | 74.97 | 73.77* | 81.55v | 83.19v | 79.71v | 80.46v |
| Wine | 97.47 | 97.47 | 97.14 | 96.40 | 95.62 | 96.07 |
| Wisconsin | 96.38 | 96.38 | 96.69 | 97.20 | 96.00 | 96.92 |
| Summary (v//*) | | (0/13/3) | (3/12/1) | (2/7/7) | (1/8/7) | (2/10/4) |

from the data. The algorithm adopts a divide-and-conquer strategy such that it removes instances already covered by the current ruleset during the learning processing. Essentially, a rule is created by building a pruned tree for the current set of instances; the branch leading to a leaf with the highest coverage is promoted to a classification rule. In this paper, this method is empirically learned with a confident factor of 0.25.

– J48 is based on ID3 (Quinlan 1993) and creates decision trees by choosing the most informative features and recursively partitioning a training data table into subtables based on the values of such features. Each node in the tree represents a feature, with the subsequent nodes branching from the possible values of this node according to the current subtable. Partitioning stops when all data items in the subtable have the same classification. A leaf node is then created to represent this classification. In this paper, J48 is set with the pruning confidence threshold $C = 0.25$.

– SMO (Smola and Schölkopf 1998) is an algorithm for efficiently solving optimisation problems which arise during the training of a support vector machine (Cortes and Vapnik 1995). It breaks optimisation problems into a series of smallest possible subproblems, which are then resolved analytically. In this paper, SMO is set with $C = 1$, tolerance $L = 0.001$, round-off error=$10^{-12}$, data running on normalised and polynomial kernel.

– NB (Naive Bayes) (John and Langley 1995) is a simple probabilistic classifier, directly applying Bayes' theorem (Papoulis 1984) with strong (naive) independence assumptions. Depending on the precise nature of the

**Table 7** Classification accuracy of MFNN_G

| Dataset | MFNN_G | PART | J48 | SMO | NB |
|---|---|---|---|---|---|
| Cleveland | 53.44 | 52.44 | 53.39 | 58.31 | 56.06 |
| Ecoli | 80.57 | 81.79 | 82.83 | 83.48 | 85.50v |
| Glass | 73.54 | 69.12 | 68.08 | 57.77* | 47.70* |
| Handwritten | 91.13 | 79.34* | 76.13* | 93.58v | 86.19* |
| Heart | 76.63 | 77.33 | 78.15 | 83.89v | 83.59v |
| Liver | 62.81 | 65.25 | 65.84 | 57.98 | 54.89 |
| Multifeat | 97.57 | 94.68* | 94.62* | 98.39v | 95.27* |
| Olitos | 78.67 | 67.00* | 65.75* | 87.92v | 78.50 |
| Page-block | 96.04 | 96.93v | 96.99v | 92.84* | 90.01* |
| Satellite | 90.92 | 86.63* | 86.41* | 86.78* | 79.59* |
| Sonar | 85.25 | 77.40* | 73.61* | 76.60* | 67.71* |
| Water2 | 84.38 | 83.85 | 83.18 | 83.64 | 69.72* |
| Water3 | 79.82 | 82.72 | 81.59 | 87.21v | 85.49v |
| Waveform | 73.77 | 77.62v | 75.25 | 86.48v | 80.01v |
| Wine | 97.47 | 92.24* | 93.37 | 98.70 | 97.46 |
| Wisconsin | 96.38 | 95.68 | 95.44 | 97.01 | 96.34 |
| Summary (v//*) | | (2/8/6) | (1/10/5) | (6/6/4) | (4/5/7) |

probability model used, naive Bayesian classifiers can be trained very efficiently in a supervised learning setting. The learning only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification.

The results are listed in Tables 7, 8 and 9, together with a statistical comparison between each method and MFNN_G, MFNN_E and MFNN_A, respectively.

**Table 8** Classification accuracy of MFNN_E

| Dataset | MFNN_E | PART | J48 | SMO | NB |
|---|---|---|---|---|---|
| Cleveland | 53.64 | 52.44 | 53.39 | 58.31 | 56.06 |
| Ecoli | 81.93 | 81.79 | 82.83 | 83.48 | 85.50 |
| Glass | 74.29 | 69.12 | 68.08 | 57.77* | 47.70* |
| Handwritten | 91.20 | 79.34* | 76.13* | 93.58v | 86.19* |
| Heart | 76.70 | 77.33 | 78.15 | 83.89v | 83.59v |
| Liver | 63.07 | 65.25 | 65.84 | 57.98 | 54.89 |
| Multifeat | 97.59 | 94.68* | 94.62* | 98.39v | 95.27* |
| Olitos | 78.83 | 67.00* | 65.75* | 87.92v | 78.50 |
| Page-block | 96.18 | 96.93v | 96.99v | 92.84* | 90.01* |
| Satellite | 91.52 | 86.63* | 86.41* | 86.78* | 79.59* |
| Sonar | 85.35 | 77.40* | 73.61* | 76.60* | 67.71* |
| Water2 | 85.21 | 83.85 | 83.18 | 83.64 | 69.72* |
| Water3 | 80.77 | 82.72 | 81.59 | 87.21v | 85.49v |
| Waveform | 74.97 | 77.62v | 75.25 | 86.48v | 80.01v |
| Wine | 97.47 | 92.24* | 93.37 | 98.70 | 97.46 |
| Wisconsin | 96.38 | 95.68 | 95.44 | 97.01 | 96.34 |
| Summary | (v//*) | (2/8/6) | (1/10/5) | (6/6/4) | (3/6/7) |

**Table 9** Classification accuracy of MFNN_A

| Dataset | MFNN_A | PART | J48 | SMO | NB |
|---|---|---|---|---|---|
| Cleveland | 58.46 | 52.44* | 53.39 | 58.31 | 56.06 |
| Ecoli | 86.85 | 81.79* | 82.83* | 83.48 | 85.50 |
| Glass | 68.95 | 69.12 | 68.08 | 57.77* | 47.70* |
| Handwritten | 91.37 | 79.34* | 76.13* | 93.58v | 86.19* |
| Heart | 82.19 | 77.33 | 78.15 | 83.89 | 83.59 |
| Liver | 66.26 | 65.25 | 65.84 | 57.98* | 54.89* |
| Multifeat | 97.95 | 94.68* | 94.62* | 98.39 | 95.27* |
| Olitos | 80.75 | 67.00* | 65.75* | 87.92v | 78.50 |
| Page-block | 95.99 | 96.93v | 96.99v | 92.84* | 90.01* |
| Satellite | 90.99 | 86.63* | 86.41* | 86.78* | 79.59* |
| Sonar | 79.38 | 77.40 | 73.61 | 76.60 | 67.71* |
| Water2 | 85.15 | 83.85 | 83.18 | 83.64 | 69.72* |
| Water3 | 81.28 | 82.72 | 81.59 | 87.21v | 85.49v |
| Waveform | 81.55 | 77.62* | 75.25* | 86.48v | 80.01 |
| Wine | 97.14 | 92.24* | 93.37 | 98.70 | 97.46 |
| Wisconsin | 96.69 | 95.68 | 95.44 | 97.01 | 96.34 |
| Summary | (v//*) | (1/7/8) | (1/9/6) | (4/8/4) | (1/7/8) |

It can be seen from these results that in general, all three implemented MFNN methods perform well. In particular, even considering MFNN_G, the least performer amongst the three, for the *glass*, *satellite*, *sonar*, and *water 2* datasets, it achieves statistically better classification performance against all the other types of classifier. Only for the *ecoli* and *waveform* datasets, MFNN_G does not perform so well as it does on the other datasets.

Amongst the three MFNN implementations, MFNN_A is again the best performer. It is able to generally improve the classification accuracies achievable by both MFNN_G and MFNN_E, for the *cleveland*, *ecoli*, *heart*, *liver*, *waveform* datasets. However, from a statistical perspective, its performance is similar to the other two overall. Nevertheless, in terms of accuracy, MFNN_A is statistically better than PART, J48 and NB, respectively, for 8, 6 and 8 datasets, with an equal statistical performance to that of SMO.

## 5.5 Comparison with the state of the art: use of different similarity metrics

It was mentioned previously that the MFNN approach offers significant flexibility as it allows the use of different similarity metrics and aggregation operators. To provide a more comprehensive view of the performance of MFNN, this section investigates the effect of employing different similarity metrics. In particular, kernel-based fuzzy similarity metrics are employed in this section (Qu et al. 2015). Such similarity metrics are induced by the stationary kernel functions and robustness in statistics. Specifically, as the overall best, MFNN_A is modified to use either the wave kernel function or the rational quadratic kernel function. The resultant classifiers are denoted by MFNN_AW and MFNN_AR, respectively.

As demonstrated in Tables 10 and 11, the classification accuracies achieved by MFNN_AW and MFNN_AR further improve over those achieved by MFNN_A for a number of datasets. Statistically, it is shown that MFNN_AW generally

**Table 10** Classification accuracy of MFNN_AW

| Dataset | MFNN_AW | PART | J48 | SMO | NB |
|---|---|---|---|---|---|
| Cleveland | 58.46 | 52.44* | 53.39 | 58.31 | 56.06 |
| Ecoli | 87.12 | 81.79* | 82.83* | 83.48 | 85.50 |
| Glass | 66.76 | 69.12 | 68.08 | 57.77* | 47.70* |
| Handwritten | 91.32 | 79.34* | 76.13* | 93.58v | 86.19* |
| Heart | 82.78 | 77.33 | 78.15 | 83.89 | 83.59 |
| Liver | 64.65 | 65.25 | 65.84 | 57.98* | 54.89* |
| Multifeat | 98.05 | 94.68* | 94.62* | 98.39 | 95.27* |
| Olitos | 82.42 | 67.00* | 65.75* | 87.92 | 78.50 |
| Page-block | 96.40 | 96.93v | 96.99v | 92.84* | 90.01* |
| Satellite | 90.76 | 86.63* | 86.41* | 86.78* | 79.59* |
| Sonar | 81.31 | 77.40 | 73.61* | 76.60 | 67.71* |
| Water2 | 85.90 | 83.85 | 83.18 | 83.64 | 69.72* |
| Water3 | 81.15 | 82.72 | 81.59 | 87.21v | 85.49v |
| Waveform | 81.09 | 77.62* | 75.25* | 86.48v | 80.01 |
| Wine | 96.53 | 92.24 | 93.37 | 98.70 | 97.46 |
| Wisconsin | 96.66 | 95.68 | 95.44 | 97.01 | 96.34 |
| Summary | (v//*) | (1/8/7) | (1/8/7) | (3/9/4) | (1/7/8) |

**Table 11** Classification accuracy of MFNN_AR

| Dataset | MFNN_AR | PART | J48 | SMO | NB |
|---|---|---|---|---|---|
| Cleveland | 57.21 | 52.44 | 53.39 | 58.31 | 56.06 |
| Ecoli | 87.68 | 81.79* | 82.83* | 83.48* | 85.50 |
| Glass | 74.98 | 69.12 | 68.08 | 57.77* | 47.70* |
| Handwritten | 91.29 | 79.34* | 76.13* | 93.58v | 86.19* |
| Heart | 81.93 | 77.33 | 78.15 | 83.89 | 83.59 |
| Liver | 64.88 | 65.25 | 65.84 | 57.98* | 54.89* |
| Multifeat | 97.75 | 94.68* | 94.62* | 98.39v | 95.27* |
| Olitos | 80.67 | 67.00* | 65.75* | 87.92v | 78.50 |
| Page-block | 96.09 | 96.93v | 96.99v | 92.84* | 90.01* |
| Satellite | 90.01 | 86.63* | 86.41* | 86.78* | 79.59* |
| Sonar | 77.02 | 77.40 | 73.61 | 76.60 | 67.71* |
| Water2 | 85.03 | 83.85 | 83.18 | 83.64 | 69.72* |
| Water3 | 80.82 | 82.72 | 81.59 | 87.21v | 85.49v |
| Waveform | 80.14 | 77.62* | 75.25* | 86.48v | 80.01 |
| Wine | 95.97 | 92.24 | 93.37 | 98.70 | 97.46 |
| Wisconsin | 96.35 | 95.68 | 95.44 | 97.01 | 96.34 |
| Summary | (v//*) | (1/9/6) | (1/9/6) | (5/6/5) | (1/7/8) |

outperforms the other approaches (particularly for the *olitos* dataset). MFNN_AR also outperforms all the other classifiers with the exception of SMO (to which MFNN_AR obtains a comparable and statistically equal performance). Note that for the *glass* dataset, MFNN_AR achieves a very significant improvement in accuracy.

In summary, examining all of the results obtained, it has been experimentally shown that when a kernel-based fuzzy similarity metric is employed, MFNN offers a better and more robust performance than the other classifiers.

## 6 Conclusion

This paper has presented a multi-functional nearest-neighbour approach (MFNN). In this work, the combination of fuzzy similarities and class memberships may be performed using different aggregators. Such an aggregated similarity measure is then employed as the decision qualifier for the process of decision-making in classification.

The wide and variable choice of the aggregators and fuzzy similarity metrics ensures that the proposed approach has a high flexibility and is of significant generality. For example, using appropriate similarity relations, aggregators and class membership functions, MFNN can perform the tasks of classical *k*NN and FNN. Such construction helps to ensure that the resulting MFNN is adaptive in dealing with different classification problems given a wide range of choices. Furthermore, using the *Gödel S*-norm and *Addition* operator, the resulting specific MFNN implementations (with crisp

membership) have the ability to achieve the same classification accuracy as two advanced fuzzy-rough set methods: fuzzy-rough nearest neighbour (FRNN) and vaguely quantified nearest neighbours (VQNN). That is, both traditional nearest-neighbour methods and advanced fuzzy-rough-based classifiers can be seen as special cases of MFNN. This observation indicates that the MFNN algorithm grants a flexible framework to the existing nearest-neighbour classification methods. These results are proven by theoretical analysis, supported with empirical results.

To demonstrate the efficacy of the MFNN approach, systematic experiments have been carried out from the perspective of classification accuracy. The results of the experimental evaluation have been very promising. They demonstrate that implemented with specific aggregators, particularly whilst employing the *Addition* operator, MFNN can generally outperform a range of state-of-the-art learning classifiers in terms of these performance indicators.

Topics for further research include a more comprehensive study of how the proposed approach would perform in regression or other prediction tasks, where the decision variables are not crisp. Also, recently, a proposal has been made to develop techniques for efficient information aggregation and unsupervised feature selection, which exploits the concept of nearest-neighbour-based data reliability (Boongoen and Shen 2010; Boongoen et al. 2011). An investigation into how the present work could be used to perform such tasks or perhaps (semi-)supervised feature selection (Jensen and Shen 2008) remains active research.

**Compliance with ethical standards**

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

Armanino C, Leardia R, Lanteria S, Modi G (1989) Chemometric analysis of tuscan olive oils. Chemom Intell Lab Syst 5:343–354

Baets BD, Mesiar R (1998) T-partitions. Fuzzy Sets Syst 97:211–223

Baets BD, Mesiar R (2002) Metrics and T-equalities. J Math Anal Appl 267:531–547

Bengio Y, Grandvalet Y (2005) Bias in estimating the variance of $K$-fold cross-validation. In: Duchesne P, RÉMillard B (eds) Statistical modeling and analysis for complex data problems. Springer, Baston, pp 75–95

Blake CL, Merz CJ (1998) UCI repository of machine learning databases. University of California, School of Information and Computer Sciences, Irvine

Boongoen T, Shang C, Iam-On N, Shen Q (2011) Nearest-neighbor guided evaluation of data reliability and its applications. IEEE Trans Syst Man Cybern Part B Cybern 41:1705–1714

Borkowski L (ed) (1970) Selected works by Jan Łukasiewicz. North-Holland Publishing Co., Amsterdam

Boongoen T, Shen Q (2010) Nearest-neighbor guided evaluation of data reliability and its applications. IEEE Trans Syst Man Cybern Part B Cybern 40:1622–1633

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth & Brooks, Monterey, CA

Cornelis C, Cock MD, Radzikowska A (2007) Vaguely quantified rough sets. In: Lecture notes in artificial intelligence, vol 4482. pp 87–94

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297

Cover T, Hart P (1967) Nearest neighbor pattern classification. IEEE Trans. Inf. Theory 13:21–27

Daelemans W, den Bosch AV (2005) Memory-based language processing. Cambridge University Press, Cambridge

Das M, Chakraborty MK, Ghoshal TK (1998) Fuzzy tolerance relation, fuzzy tolerance space and basis. Fuzzy Sets Syst 97:361–369

Dienes SP (1949) On an implication function in many-valued systems of logic. J Symb Logic 14:95–97

Dubois D, Prade H (1992) Putting rough sets and fuzzy sets together. Intelligent decision support, Springer, Dordrecht, pp 203–232

Duda RO, Hart PE, Stork DG (2001) Pattern classification, 2nd edn. Wiley, Hoboken

Jensen R, Cornelis C (2011) A new approach to fuzzy-rough nearest neighbour classification. In: Transactions on rough sets XIII, LNCS, vol 6499. pp 56–72

Jensen R, Shen Q (2008) Computational intelligence and feature selection: rough and fuzzy approaches. Wiley, Indianapolis

Jensen R, Shen Q (2009) New approaches to fuzzy-rough feature selection. IEEE Trans Fuzzy Syst 17:824–838

John GH, Langley P (1995) Estimating continuous distributions in Bayesian classifiers. In: Eleventh conference on uncertainty in artificial intelligence, pp 338–345

Keller JM, Gray MR, Givens JA (1985) A fuzzy k-nearest neighbor algorithm. IEEE Trans Syst Man Cybern 15:580–585

Kleene SC (1952) Introduction to metamathematics. Van Nostrand, New York

Kolmogorov AN (1950) Foundations of the theory of probability. Chelsea Publishing Co., Chelsea

Papoulis A (1984) Probability, random variables, and stochastic processes, 2nd edn. McGraw-Hill, New York

Pawlak Z (1991) Rough sets: theoretical aspects of reasoning about data. Kluwer Academic Publishing, Norwell

Qu Y, Shang C, Shen Q, Mac Parthaláin N, Wu W (2015) Kernel-based fuzzy-rough nearest-neighbour classification for mammographic risk analysis. Int J Fuzzy Syst 17:471–483

Quinlan JR (1993) C4.5: Programs for machine learning., The Morgan Kaufmann series in machine learningMorgan Kaufmann, Burlington

Radzikowska AM, Kerre EE (2002) A comparative study of fuzzy rough sets. Fuzzy Sets Syst 126:137–155

Sarkar M (2007) Fuzzy-rough nearest neighbors algorithm. Fuzzy Sets Syst 158:2123–2152

Smola AJ, Schölkopf B (1998) A tutorial on support vector regression. Neuro-COLT2 Technical Report Series, NC2-TR-1998-030

Witten IH, Frank E (1998) Generating accurate rule sets without global optimisation. In: Proceedings of the 15th international conference on machine learning, San Francisco. Morgan Kaufmann

Witten IH, Frank E (2000) Data mining: practical machine learning tools with Java implementations. Morgan Kaufmann, Burlington

Yager RR (1988) On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Trans Syst Man Cybern 18:183–190

Yao YY (1998) A comparative study of fuzzy sets and rough sets. Inf Sci 109:227–242