

A New Approach to Fuzzy-Rough Nearest Neighbour Classification

Richard Jensen¹ and Chris Cornelis²

¹ Dept. of Comp. Sci., Aberystwyth University, Ceredigion, SY23 3DB, Wales, UK
rkj@aber.ac.uk

² Dept. of Appl. Math. and Comp. Sci., Ghent University, Gent, Belgium
Chris.Cornelis@UGent.be

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

1 Introduction

The K -nearest neighbour (KNN) algorithm [7] 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 [11]. 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 [14], the FNN algorithm has problems dealing adequately with insufficient knowledge. In particular, when every training pattern is far removed from the test object, and hence there are no suitable neighbours, the algorithm is still forced to make clear-cut predictions. This is because the predicted membership degrees to the various decision classes always need to sum up to 1.

To address this problem, Sarkar [14] introduced a so-called fuzzy-rough ownership function that, when plugged into the conventional FNN algorithm, produces class confidence values that do not necessarily sum up to 1. However, this method (called FRNN-O throughout this paper) does not refer to the main ingredients of rough set theory, i.e., lower and upper approximation. In this paper, therefore, we present an alternative approach, which uses a test object’s nearest neighbours to construct the lower and upper approximation of each decision

class, and then computes the membership of the test object to these approximations. The method is very flexible, as there are many options to define the fuzzy-rough approximations, including the traditional implicator/t-norm based model [13], as well as the vaguely quantified rough set (VQRS) model [3], which is more robust in the presence of noisy data.

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

2 Hybridization of Rough Sets and Fuzzy Sets

Rough set theory (RST) [12] provides a tool by which knowledge may be extracted from a domain in a concise way; it is able to retain the information content whilst reducing the amount of knowledge involved. Central to RST is the concept of indiscernibility. Let (\mathbb{U}, \mathbb{A}) be an information system³, where \mathbb{U} is a non-empty set of finite objects (the universe of discourse) and \mathbb{A} is a non-empty finite set of attributes such that $a : \mathbb{U} \rightarrow V_a$ for every $a \in \mathbb{A}$. V_a is the set of values that attribute a may take. With any $B \subseteq \mathbb{A}$ there is an associated equivalence relation R_B :

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

If $(x, y) \in R_B$, then x and y are indiscernible by attributes from B . The equivalence classes of the B -indiscernibility relation are denoted $[x]_B$. Let $A \subseteq \mathbb{U}$. A can be approximated using the information contained within B by constructing the B -lower and B -upper approximations of A :

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

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

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

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

³ In the classification problems considered further on in this paper, $\mathbb{A} = \mathbb{C} \cup \{d\}$, where \mathbb{C} represents the set of conditional attributes, and d is the decision or class attribute.

Given a fuzzy tolerance relation R and a fuzzy set A in \mathbb{U} , the lower and upper approximation of A by R can be constructed in several ways. A general definition [4, 13] is the following:

$$(R\downarrow A)(x) = \inf_{y \in \mathbb{U}} I(R(x, y), A(y)) \quad (4)$$

$$(R\uparrow A)(x) = \sup_{y \in \mathbb{U}} T(R(x, y), A(y)) \quad (5)$$

Here, I is an implicator⁴ and T a t-norm⁵. When A is a crisp (classical) set and R is an equivalence relation in \mathbb{U} , the traditional lower and upper approximation are recovered.

Just like their crisp counterparts, formulas (4) and (5) (henceforth called the FRS approximations) are quite sensitive to noisy values. That is, a change in a single object can result in drastic changes to the approximations (due to the use of sup and inf, which generalize the existential and universal quantifier, respectively). In the context of classification tasks, this behaviour may affect accuracy adversely. Therefore, in [3], the concept of vaguely quantified rough sets (VQRS) was introduced. It uses the linguistic quantifiers “most” and “some”, as opposed to the traditionally used crisp quantifiers “all” and “at least one”, to decide to what extent an object belongs to the lower and upper approximation. Given a couple (Q_u, Q_l) of fuzzy quantifiers⁶ that model “most” and “some”, the lower and upper approximation of A by R are defined by

$$(R\downarrow A)(y) = Q_u \left(\frac{|Ry \cap A|}{|Ry|} \right) = Q_u \left(\frac{\sum_{x \in X} \min(R(x, y), A(x))}{\sum_{x \in X} R(x, y)} \right) \quad (6)$$

$$(R\uparrow A)(y) = Q_l \left(\frac{|Ry \cap A|}{|Ry|} \right) = Q_l \left(\frac{\sum_{x \in X} \min(R(x, y), A(x))}{\sum_{x \in X} R(x, y)} \right) \quad (7)$$

where the fuzzy set intersection is defined by the min t-norm and the fuzzy set cardinality by the sigma-count operation. As an important difference to (4) and (5), the VQRS approximations do not extend the classical rough set approximations, in a sense that when A and R are crisp, $R\downarrow A$ and $R\uparrow A$ may still be fuzzy.

⁴ An implicator I is a $[0, 1]^2 \rightarrow [0, 1]$ mapping that is decreasing in its first and increasing in its second argument, satisfying $I(0, 0) = I(0, 1) = I(1, 1) = 1$ and $I(1, 0) = 0$.

⁵ A t-norm T is an increasing, commutative, associative $[0, 1]^2 \rightarrow [0, 1]$ mapping satisfying $T(x, 1) = x$ for x in $[0, 1]$.

⁶ By a fuzzy quantifier, we mean an increasing $[0, 1] \rightarrow [0, 1]$ mapping such that $Q(0) = 0$ and $Q(1) = 1$.

3 Fuzzy Nearest Neighbour Classification

The fuzzy K -nearest neighbour (FNN) algorithm [11] 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:

$$C(y) = \sum_{x \in N} R(x, y)C(x) \quad (8)$$

where N is the set of object y 's K nearest neighbours, and $R(x, y)$ is the $[0,1]$ -valued similarity of x and y . In the traditional approach, this is defined in the following way:

$$R(x, y) = \frac{\|y - x\|^{-2/(m-1)}}{\sum_{j \in N} \|y - j\|^{-2/(m-1)}} \quad (9)$$

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

FNN($\mathbb{U}, \mathcal{C}, y, K$).

\mathbb{U} , the training data; \mathcal{C} , the set of decision classes;

y , the object to be classified; K , the number of nearest neighbours.

- (1) $N \leftarrow \text{getNearestNeighbours}(y, K)$;
- (2) $\forall C \in \mathcal{C}$
- (3) $C(y) = \sum_{x \in N} R(x, y)C(x)$
- (4) **output** $\arg \max_{C \in \mathcal{C}} (C(y))$

Fig. 1. The fuzzy KNN algorithm

Initial attempts to combine the FNN algorithm with concepts from fuzzy rough set theory were presented in [14, 16]. 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 ,

$$\tau_C(y) = \frac{\sum_{x \in \mathbb{U}} R(x, y)C(x)}{|\mathbb{U}|} \quad (10)$$

In this, the fuzzy relation R is determined by:

$$R(x, y) = \exp \left(- \sum_{a \in \mathbb{C}} \kappa_a (a(y) - a(x))^{2/(m-1)} \right) \quad (11)$$

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

$$\kappa_a = \frac{|\mathbb{U}|}{2 \sum_{x \in \mathbb{U}} \|a(y) - a(x)\|^{2/(m-1)}} \quad (12)$$

$\tau_C(y)$ 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 Figure 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 (10). Finally, when all training objects have been considered, the algorithm outputs the class with highest membership. The algorithm's complexity is $O(|\mathbb{C}||\mathbb{U}| + |\mathbb{U}| \cdot (|\mathbb{C}| + |C|))$.

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 $N \leftarrow \text{getNearestNeighbours}(y, K)$.

FRNN-O($\mathbb{U}, \mathbb{C}, \mathcal{C}, y$).

\mathbb{U} , the training data; \mathbb{C} , the set of conditional features;

\mathcal{C} , the set of decision classes; y , the object to be classified.

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

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

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 [1]. However, the authors did not state how to use the upper and lower approximations to derive classifications.

4 Fuzzy-Rough Nearest Neighbour (FRNN) Algorithm

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

In particular, if $(R\downarrow C)(y)$ is high, it reflects that all (most) of y 's neighbours belong to C , while a high value of $(R\uparrow C)(y)$ means that at least one (some) neighbour(s) belong(s) to that class, depending on whether the FRS or VQRS approximations are used. To perform crisp classification, the algorithm outputs the decision class with the resulting best fuzzy lower and upper approximation memberships. The complexity of the algorithm is $O(|\mathcal{C}| \cdot (2|\mathbb{U}|))$.

FRNN($\mathbb{U}, \mathcal{C}, y$).

\mathbb{U} , the training data; \mathcal{C} , the set of decision classes;

y , the object to be classified.

```

(1)    $N \leftarrow \text{getNearestNeighbors}(y, K)$ 
(2)    $\mu_1(y) \leftarrow 0, \mu_2(y) \leftarrow 0, \text{Class} \leftarrow \emptyset$ 
(3)    $\forall C \in \mathcal{C}$ 
(4)     if  $((R\downarrow C)(y) \geq \mu_1(y) \ \&\& \ (R\uparrow C)(y) \geq \mu_2(y))$ 
(5)        $\text{Class} \leftarrow C$ 
(6)        $\mu_1(y) \leftarrow (R\downarrow C)(y), \mu_2(y) \leftarrow (R\uparrow C)(y)$ 
(7)   output  $\text{Class}$ 

```

Fig. 3. The fuzzy-rough nearest neighbour algorithm

When using FRNN-FRS, the use of K is not required in principle: as $R(x, y)$ gets smaller, x tends to have only a minor influence on $(R\downarrow C)(y)$ and $(R\uparrow C)(y)$. For FRNN-VQRS, this may generally not be true, because $R(x, y)$ appears in the numerator as well as the denominator of (6) and (7).

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

$$R(x, y) = \min_{a \in \mathcal{C}} R_a(x, y) \quad (13)$$

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

$$R_a^1(x, y) = \exp\left(-\frac{(a(x) - a(y))^2}{2\sigma_a^2}\right) \quad (14)$$

$$R_a^2(x, y) = 1 - \frac{|a(x) - a(y)|}{|a_{\max} - a_{\min}|} \quad (15)$$

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

5 Experimentation

This section presents the initial experimental evaluation of the classification methods FNN, FRNN-O, FRNN-FRS and FRNN-VQRS for the task of pattern classification, over nine benchmark datasets from [2] and [14]. The details of the datasets used can be found in table 1. All of them have a crisp decision attribute.

Dataset	Objects	Attributes
Cleveland	297	14
Glass	214	10
Heart	270	14
Ionosphere	230	35
Letter-dgoq	3114	17
Olitos	120	26
Water 2	390	39
Water 3	390	39
Wine	178	14

Table 1. Dataset details

5.1 Experimental Setup

K is initialized as $|\mathbb{U}|$, the number of objects in the dataset and then decremented by $1/30^{th}$ of $|\mathbb{U}|$ each time, resulting in 30 experiments for each dataset. For each choice of parameter K , 2×10 -fold cross-validation is performed. For FNN and FRNN-O, m is set to 2. For the new approaches, $R = \dots$ was chosen. In the FRNN-FRS approach, we used the min t-norm and the Kleene-Dienes implicator I defined by $I(x, y) = \max(1 - x, y)$. The FRNN-VQRS approach was implemented using $Q_l = Q_{(0.1, 0.6)}$ and $Q_u = Q_{(0.2, 1.0)}$, according to the general formula

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

5.2 Comparative Investigation

The results of the experiments are shown in Figure 4. Several interesting observations can be made from them. First, for all but one dataset (**Letter-dgoq**, which was used in [14]), either FRNN-FRS or FRNN-VQRS yields the best results. Overall, FRNN-FRS produces the most consistent results. This is particularly remarkable considering the inherent simplicity of the method. FRNN-VQRS is best for **cleveland** and **heart**, which might be attributed to the comparative presence of noise in those datasets, but it performs rather disappointing for a number of other datasets (**glass**, **letter-dgoq**, **wine**, **olitos**).

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

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

6 Conclusion and Future Work

This paper has presented two new techniques for fuzzy-rough classification based on the use of lower and upper approximations w.r.t. fuzzy tolerance relations. The difference between them is in the definition of the approximations: while FRNN-FRS uses “traditional” operations based on a t-norm and an implicator, FRNN-VQRS uses a fuzzy quantifier-based approach. The results show that these methods are effective, and that they are competitive with existing methods such as the fuzzy K -nearest neighbour and the fuzzy-rough ownership function approach. Further investigation, however, is still needed, to adequately explain the impact of the choice of fuzzy relations, connectives and quantifiers.

Also, the impact of a feature selection preprocessing step upon classification accuracy needs to be investigated. It is expected that feature selectors that incorporate fuzzy relations expressing closeness of objects (see e.g. [5, 10]) should be able to further improve the effectiveness of the classification methods presented here.

Finally, an important challenge is to adapt the algorithms so that they can deal with continuous decision attributes. In this case, we need to predict the membership of a test object to different, possibly overlapping classes. Such a prediction can be based on the test object's membership degrees to the lower and/or upper approximation (e.g., on the average of these two values).

Acknowledgment

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

References

1. H. Bian and L. Mazlack, "Fuzzy-Rough Nearest-Neighbor Classification Approach," Proceeding of the 22nd International Conference of the North American Fuzzy Information Processing Society (NAFIPS), pp. 500–505, 2003.
2. C.L. Blake and C.J. Merz. UCI Repository of machine learning databases. Irvine, University of California, 1998. <http://archive.ics.uci.edu/ml/>
3. C. Cornelis, M. De Cock, A.M. Radzikowska, Vaguely Quantified Rough Sets, Proceedings of 11th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC2007), Lecture Notes in Artificial Intelligence 4482, pp. 87-94, 2007.
4. C. Cornelis, M. De Cock, A.M. Radzikowska, Fuzzy Rough Sets: from Theory into Practice, Handbook of Granular Computing (W. Pedrycz, A. Skowron, V. Kreinovich, eds.), Wiley, 2008.
5. C. Cornelis, G. Hurtado Martín, R. Jensen, D. Slezak, Feature Selection with Fuzzy Decision Reducts, Proceedings of 3rd International Conference on Rough Sets and Knowledge Technology (RSKT2008), 2008.
6. *Pattern Classification and Scene Analysis*, R. Duda and P. Hart, Wiley, New York, 1973.
7. R. Jensen and Q. Shen, "Fuzzy-Rough Sets Assisted Attribute Selection," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 1, pp. 73–89, 2007.
8. 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.
9. Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishing, 1991.
10. A.M. Radzikowska and E.E. Kerre, "A comparative study of fuzzy rough sets," *Fuzzy Sets and Systems*, vol. 126, no. 2, pp. 137–155, 2002.
11. M. Sarkar, "Fuzzy-Rough nearest neighbors algorithm," *Fuzzy Sets and Systems*, vol. 158, pp. 2123–2152, 2007.
12. 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.

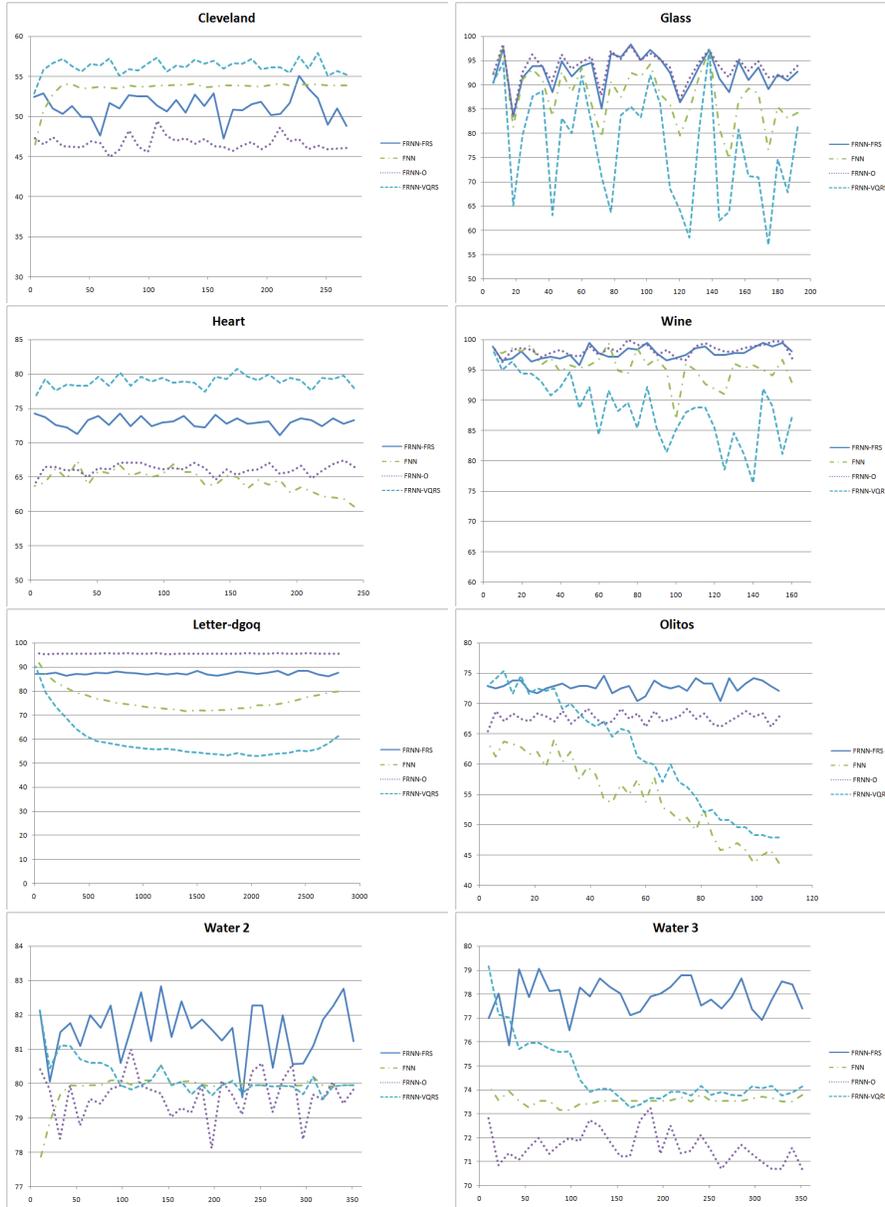


Fig. 4. Classification accuracy for the four methods and different values of K