Effect of feature weighting in Fuzzy-Rough nearest neighbourclassification algorithms 2017

Verbeke Bram BCVERBEK.VERBEKE@UGENT.BE
Dept. of Appl. Math. and Comp. Sci., Ghent University, Gent, Belgium

Miclotte Victor VICTOR.MICLOTTE@UGENT.BE
Dept. of Appl. Math. and Comp. Sci., Ghent University, Gent, Belgium

Abstract

In this paper, we discuss the effect of feature weighting methods on a nearest neighbour classification algorithm based on the Vaguely Quantified Rough Set (VQRS) model. It is shown experimentally that feature weights can improve the performance of this classification algorithm. Multiple feature weighting methods should be considered, as there are none that consistently outperform the others.

1. Introduction

When dealing with data, there is the unavoidable property that data is never perfect. Data sets contain both vague and incomplete information. An example of vagueness is the problem to describe a subjective concept such as *large, beautiful, late*. . . Incomplete information occurs when two different objects are indistinguishable according to the features at hand, yet labeled differently.

In this paper we will discuss fuzzy set theory (Zadeh, 1965), as well as rough set theory (Pawlak, 1982). Both notions aim to different purposes. Fuzzy set theory models vagueness. An object is given a membership degree to express to what extent it satisfies a certain property. Rough set theory is a way to tackle the incompleteness of data. Dubois and Prade (Dubois & Prade, 1990) were the first to propose to combine these two theories rather than to combine these two on the same problems. Now it is included in several machine learning tools.

The K-nearest Neighbour algorithm (Keller et al., 1985) is a well-known technique to classify an object x. At first, it calculates the K nearest neighbours of the instance x (these are the K objects closest to x), then the new object x will be classified by the class label of the most observed class among these K neighbours. This rule was later extended by a fuzzy variant. This would calculate the probability that an object x is included in class C. So it would take into account that an object can partially belong to several different classes.

Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

In this paper we use a different approach (FRNN), presented in (Jensen & Cornelis, 2011b), which uses the nearest neighbours of a test object x to construct the lower and upper approximation of a decision class. FRNN computes the membership degree of x to each of these approximations. In this paper we will use the Vaguely Quantified Rough Set model (VQRS) (Cornelis et al., 2007) claiming this model is more robust to make classification errors. In this model the approximations are constructed using fuzzy quantifiers.
The remainder of this paper is structured as follows. In section 2 we give some preliminaries we use in the paper. In section 3 the algorithms are explained. Section 4 gives a survey of the weighting methods we used. Section 5 explains our experiments.

2. Preliminaries

In this section we discuss fuzzy set theory, rough set theory and how these can be combined.

2.1. Fuzzy Sets

Fuzzy set theory (Zadeh, 1965) allows that objects belong to a set to a given degree, which extends the notion of a classical set. A fuzzy set is any mapping $A : X \to [0, 1]$. If X is finite, the cardinality of the fuzzy set, is defined by

$$|A| = \sum_{x \in X} A(x),$$

where $A(x)$ is the membership degree of an element x to the fuzzy set A.

Not only the classical notion of a set is extended, analogously we can extend the notion of a relation. Where the former notion of a relation between elements from X and Y is known as a subset of $X \times Y$, we know for a fuzzy relation

$$R : X \times Y \to [0,1]$$

that if $\forall x \in X : R(x,x) = 1$, R is reflexive and if $\forall x,y \in X : R(x,y) = R(y,x)$, R is symmetric. If R fulfills both conditions it is called a fuzzy tolerance relation.

The traditional concepts of conjunction and implication can be extended to a fuzzy setting. The following extension, named triangular norm (t-norm for short) is any mapping $T : [0,1]^2 \to [0,1]$ satisfying the following four conditions.

1. $(\forall x \in [0,1])(T(x,1) = x)$
2. $(\forall x,y \in [0,1]^2)(T(x,y) = T(y,x))$
3. $(\forall x,y,z \in [0,1]^3)(T(x,T(y,z)) = T(T(x,y),z))$
4. $(\forall x_1,x_2,y_1,y_2 \in [0,1]^4)$ $(x_1 \leq x_2 \wedge y_1 \leq y_2 \Rightarrow T(x_1,y_1) \leq T(x_2,y_2))$

A commonly used t-norm is the minimum operator. Another extension to the fuzzy setting is any mapping $I : [0,1]^2 \to [0,1]$ satisfying the following four conditions. It is called an implicator.

1. $(\forall x_1,x_2,y \in [0,1])(x_1 \leq x_2 \Rightarrow I(x_1,y) \geq I(x_2,y))$
2. $(\forall x,y_1,y_2 \in [0,1])(y_1 \leq y_2 \Rightarrow I(x,y_1) \leq I(x,y_2))$
3. $I(0,0) = I(0,1) = I(1,1) = 1$ en $I(1,0) = 0$

An example of an implicator is the Kleenes-Dienes implicator I_M defined by $I_M(x,y) = \max(1-x,y)$

2.2. Rough Sets

Let us observe our data as an information system (X,A) where A is a non-empty finite set of attributes (features) and X is a non-empty set of objects. Every $a \in A$ is a fuzzy set, where we map $a : X \to [0,1]$ every object $x \in X$ to his membership degree of a, $a(x)$. With any subset $B \subseteq A$ we can associate a strict relation

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

The theory about rough sets handles incomplete data. The problem with incomplete data is that we can observe two elements x,y which we cannot discern based on our features ($(x,y) \in R_A$), though the two are classified with two different class labels. According to these features we have no clue why these objects would not belong to the same class.

The purpose of rough set theory (Pawlak, 1982) is to make a lower ($R_B \downarrow A$) and upper approximation ($R_B \uparrow A$) of a set, with

$$R_B \downarrow A = \{x \in X | \, [x]_B \subseteq A\}$$
$$R_B \uparrow A = \{x \in X | \, [x]_B \cap A \neq \emptyset\}$$

A decision system $(X, A \cup d)$ is a special kind of information system, where it is known in advance that the conditional features are included in A and the decision attribute will be $d \notin A$.

2.3. Fuzzy-Rough Set Theory

Given a fuzzy tolerance relation R and a fuzzy set A in X, the lower and upper approximations of A by R can be constructed in several ways. A general definition for a fuzzy-rough set was given by Radzikowska and Kerre (Radzikowska & Kerre, 2002).

$$(R \downarrow A)(x) = \inf_{x \in X} \{I(R(x,y), A(y))\} \quad (1)$$

$$(R \uparrow A)(x) = \sup_{x \in X} \{T(R(x,y), A(y))\} \quad (2)$$

where we use a t-norm T and an implicator I. We remark this method is very flexible according to the facility to choose T and I. Due to the use of the infimum and supremum, these formulas (1), (2) are very sensitive to noisy values. This could influence our classification outcome very badly.
Therefore the concept of vaguely quantified rough sets was introduced in (Cornelis et al., 2007). It reflects the use of the crisp quantifiers $\forall$ and $\exists$ and they replace them by the linguistic quantifiers 'most' and 'some', which we implement as fuzzy quantifiers.
A fuzzy quantifier is any increasing $[0,1] \to [0,1]$ mapping such that $Q(0) = 0$ and $Q(1) = 1$. Examples of fuzzy quantifiers can be made using the following formula, for $0 \leq \alpha < \beta \leq 1$ and $x \in [0,1]$,

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

In this paper, $Q_{(0.1,0.6)}$ and $Q_{(0.2,1)}$ are used respectively to reflect the vague quantifiers 'some' and 'most'. These two quantifiers can be used in another model of a fuzzy-rough set.
Given a couple (Q_l, Q_u) of fuzzy quantifiers, the lower and upper approximation of A by R are defined by

$$(R \downarrow A)(x) = Q_l \left(\frac{\sum_{y \in X} \min(R(x,y), A(y))}{\sum_{y \in X} \min R(x,y)} \right) \quad (3)$$

$$(R \uparrow A)(x) = Q_u \left(\frac{\sum_{y \in X} \min(R(x,y), A(y))}{\sum_{y \in X} \min R(x,y)} \right) \quad (4)$$

3. Fuzzy-Rough Nearest Neighbours (FRNN)

In this section we explain the algorithm that performed our classification. FRNN is based on the fuzzy-rough set theory and was proposed by Jensen and Cornelis in (Jensen & Cornelis, 2011a). This algorithm computes the K nearest neighbours to construct a lower and upper approximation of each decision class. When a test object has to be classified, it calculates its membership degree in each of these approximations, the decision class with the highest degree will be assigned the class label of this test object.

Algorithm 1 AverageClassifier

```
**Input:** $X$: training data, $\mathcal{C}$: set of decision classes, $y$: the object to be classified
**Output:** Class label for $y$
$N \leftarrow getNearestNeighbours(y, K)$
$\tau \leftarrow 0, Class \leftarrow \emptyset$
**for all** $C \in \mathcal{C}$ **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 if**
**end for**
**output** $Class$
```

Algorithm 2 LowClassifier

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

Algorithm 3 UpClassifier

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

Each of these algorithms needs a relation R, which we can fill in ourselves. In (Jensen & Cornelis, 2011a) the authors used the following relation

$$R(x,y) = \min_{a \in A} R_a(x,y) \quad (5)$$

where $R_a(x,y)$ is defined for each attribute as

$$R_a(x,y) = 1 - \frac{|a(x) - a(y)|}{|a_{max} - a_{min}|} \quad (6)$$

where a_{max} is the maximal occurring value of that attribute a and a_{min} analogously the minimal occurring value. The complexity of this algorithm is $O(|X| + |\mathcal{C}| \cdot K)$.

4. Feature weighting methods

In this section we propose different methods to calculate weights for features. These weights represent the importance of a certain feature for classifying instances. We then describe the effect of these different methods on the VQNN algorithm proposed by Jensen and Cornelis in (Jensen & Cornelis, 2011a).

First we discuss information gain, information gain ratio, the symmetrical uncertainty coefficient and feature weighting based on the Kullback-Leibler measure, all of these are based on entropy. Next we discuss feature weighting methods based on support vector machines, the feature selection algorithm Relief (Kira & Rendell, 1992) and the 1R (Holte, 1993) "1-rules" classification algorithm.

4.1. Entropy-based feature weighting methods

The measure of (information) entropy is a widely used measure in information theory and machine learning. It was introduced in (Shannon & Weaver, 1949) by Shannon and Weaver. Entropy is a measure of the average information content of a discrete random variable.

The entropy H of a discrete random variable X with possible values $\{x_1, \ldots, x_n\}$ and probability mass function $P(X)$ is defined as: $H(X) = E[I(X)] = E[-ln(P(X))]$. Here $I(X)$ is the information content of X as defined in (Kobayashi et al., 2007). Intuitively when a low-probability event occurs, the event carries more "information". The amount of information conveyed by each event defined by $I(x)$ is a random variable and its expected value is the entropy.

4.1.1. Information gain

The information gain calculates the difference between the entropy of a priori distribution of the class attribute C and that of a posteriori distribution of C given the distribution of an attribute A. It is defined as follows:

$$IG(C, A) = H(C) - H(C|A).$$

It represents the discriminative power of a feature A and we will therefore use this as the weight for the attribute A.

4.1.2. Information gain ratio

The information gain ratio, defined as:

$$IGR(C, A) = \frac{H(C) - H(C|A)}{H(A)},$$

is the ratio of information gain to the intrinsic information. It makes small adjustments to the information gain to reduce by taking the distribution of the attribute A into account. Information gain ratio can also be seen as a way to represent the discriminative power of a feature A and can serve as a weight for this attribute.

4.1.3. Symmetrical uncertainty coefficient

The uncertainty coefficient is defined as:

$$U(C|A) = \frac{H(C) - H(C|A)}{H(C)}.$$

Similar to the information gain ratio, it is an adjusted version of the information gain. The symmetrical uncertainty coefficient is defined as follows.

$$SymmU(C, A) = \frac{H(C)U(C|A) + H(A)U(A|C)}{H(C) + H(A)}$$

4.1.4. Kullback-Leibler measure

The last feature weighting method based on entropy is the method proposed by Lee, Gutierrez and Dou in (Lee et al., 2011). This method uses the Kullback-Leibler measure which is defined as:

$$\mathcal{KL}(C|a_{ij}) = \sum_c P(c|a_{ij}) \log \frac{P(c|a_{ij})}{P(c)}.$$

The feature weight assigned to feature i, $wavg(i)$, is the weighted average of the Kullback-Leibler measures across the feature values.

$$wavg(i) = \sum_j \frac{\#(a_{ij})}{N} \mathcal{KL}(C|a_{ij})$$

Here $\#(a_{ij})$ is the number of instances that have value a_{ij} for feature i and N is the total number of training instances. It is shown that this feature weighting method can improve the performance of naive Bayesian learning in (Lee et al., 2011).

4.2. Other feature weighting methods

In this section we discuss three other feature weighting methods based on support vector machines, the Relief algorithm for feature selection and the 1R classification algorithm.

4.2.1. Support vector machines

For this method a support vector machine is built for every pair of classes. The SVM's are built using Platt's sequential minimal optimization algorithm for training a support vector classifier (Platt, 1998). We then compute the average of the feature weights across all support vector machines for every feature. These weights are then used as weights for the VQNN classification.

4.2.2. Relief feature selection

Relief is a feature selection algorithm developed for binary-class problems introduced by Kira and Rendell (Kira & Rendell, 1992).

Algorithm 4 Relief

```
Input: $X$: data, $\mathcal{A}$: set of $N$ attributes, $\mathcal{C}$: binary set of decision classes, $M$: number of samples, $w = (0, \ldots, 0)$: $N$-dimensional vector
Output: Weight vector $w = (w_{attr1}, \ldots, w_{attrN})$
for $m = 1 : M$ do
    randomly select $x \in X$
    find $NH(x)$ and $NM(x)$
    for all $A \in \mathcal{A}$ do
        $w_A = w_A - |A(x) - A(NH(x))| + |A(x) - A(NM(x))|$
    end for
end for
output Weight vector $w$
```

The algorithm samples M instances x and computes the "nearest hit" $NH(x)$ (resp. "nearest miss" $NM(x)$), which is the nearest instance with the same (resp. a different) class label as x. The original weight vector, which is filled with zeros is gradually updated throughout the algorithm.

In this paper multi-class data sets are used, therefore we use an enhanced version Relieff, proposed by Kononenko et al. in (Kononenko et al., 1997). Relieff is also shown to be less susceptible to noise.

4.2.3. One rules

1R is a simple, yet accurate (Holte, 1993) classification algorithm, which goes as follows.

Algorithm 5 OneR

```
Input: $X$: training data, $\mathcal{A}$: set of attributes, $\mathcal{C}$: set of decision classes
Output: Optimal attribute and its rule
for all Attribute $A \in \mathcal{A}$ do
    create a rule $R_A$
    for all Value $a \in A$ do
        find the most frequent class $C$
        add rule to $R_A$: if  $A = a$ then  $\mathcal{C} = C$
    end for
    calculate the error rate of $R_A$
end for
output $A \in \mathcal{A}$ with lowest error rate for its rule $R_A$, $R_A$
```

The algorithm computes an error rate r_A for each attribute $A \in \mathcal{A}$, we use these error rates in order to compute a weight for each attribute as follows.

$$w_A = 1 - r_A$$

The higher the value of r_A, the more misclassifications happened when using only this attribute and its rule for classification. Therefore the weight w_A for this attribute will be lower, since the attribute (with its rule) makes many mistakes.

4.3. Ensemble method for feature weighting

We propose an ensemble feature weighting method, which combines all of the previous methods. This method combines the computed weight vectors as follows.

First every weight vector is re-scaled to values in $[0, 1]$. The average vector of the re-scaled weight vectors is the final weight vector for the ensemble method.

4.4. Implement weight vector

All these methods end up with a weight vector where we can find the weight for each feature. When we look back at the similarity 5 and 6, we would insert our weights

$$R_a(x, y) = 1 - w(a) \cdot \frac{|a(x) - a(y)|}{|a_{max} - a_{min}|} \tag{7}$$

where $w(a)$ is the weight that belongs to feature a. In our experiments it appeared that this similarity measure is not very influential by weights. So we tried for another similarity measure.

$$R(x, y) = 1 - \frac{||x - y||}{d_{max}}$$

where $d_{max} = \max_{x,y \in X} ||x - y||$. Here $||x - y||$ was defined as the L^2-norm

$$||x - y|| = \sqrt{\sum_{a \in A} (a(x) - a(y))^2}$$

where we could insert our weight vector like this

$$||x - y|| = \sqrt{\sum_{a \in A} (w(a) \cdot (a(x) - a(y)))^2}$$

This similarity measure appears to react more to the different weights. Therefore we fix this similarity relation through this paper.

5. Experiments

In this section, we present the methodology used in the experiments carried out. The data sets used in our experiments are available from UCI Repository (http://www.ics.uci.edu/ mlearn/MLRepository.html).
Our data sets were split up into 10 folds, for using cross-validation. These folds were constructed before all experiments and were stored into separate data files, so we would do all our experiments with the same 10 folds. The data sets can be found in table 1.

5.1. Impact of K

First we examined the impact of the number of neighbours K on classification accuracy. For every data set 10 experiments were conducted using $K = 1, \ldots, 10$. Then we computed the accuracies and balanced accuracies for every classifier we want to examine (LowClassifier, AverageClassifier, UppClassifier). The results can be seen in Figures 4 -5 -6 -7 -8 - 9.

These experiments show that for most data sets the UpClassifier loses balanced accuracy when the number of K is increasing. This is a global result across all data sets. Apparently the number of K does not influence the accuracies of Lowclassifier. Yet we see globally that $K = 3$ is the best option to reduce classification errors. So we will keep it fixed during the following experiments.

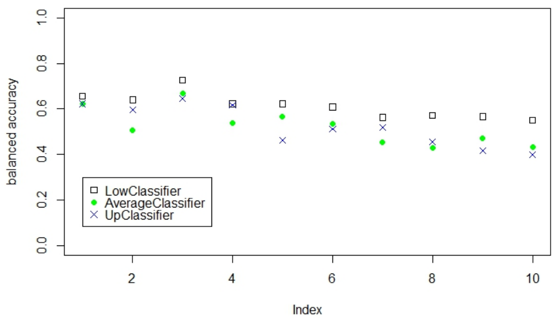

Figure 3. Impact of k, data set: Glass

6. Figures

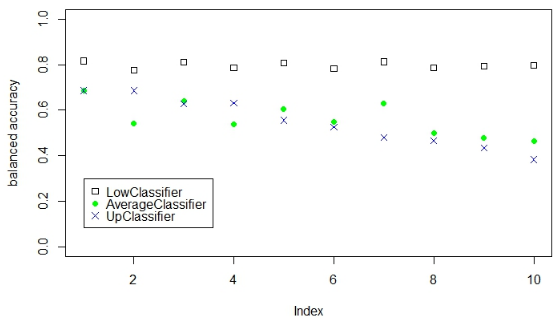

Figure 1. Impact of k, data set: Water2

Table 1. Data sets

DATA SET	NUMBER OF FEATURES	NUMBER OF INSTANCES	NUMBER OF CLASSES	TYPE
APPENDICITIS	7	106	2	NUM
GLASS	9	214	7	NUM
HEART	13	294	2	NUM
IRIS	4	150	3	NUM
WATER2	38	521	5	NUM
WINE	13	178	3	NUM

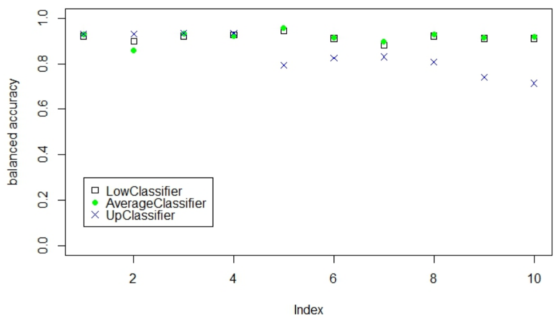

Figure 2. Impact of k, data set: Wine

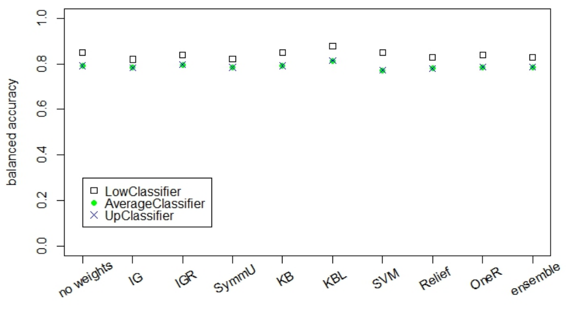

Figure 4. Balanced accuracy, data set: Appendicitis

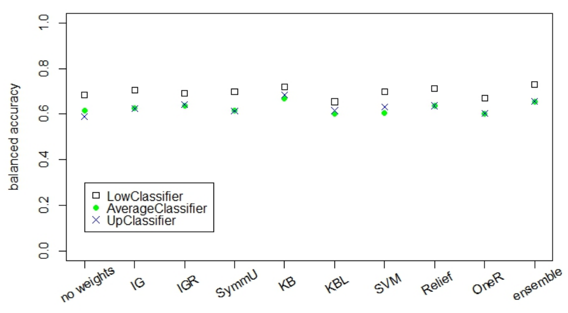

Figure 5. Balanced accuracy, data set: Glass

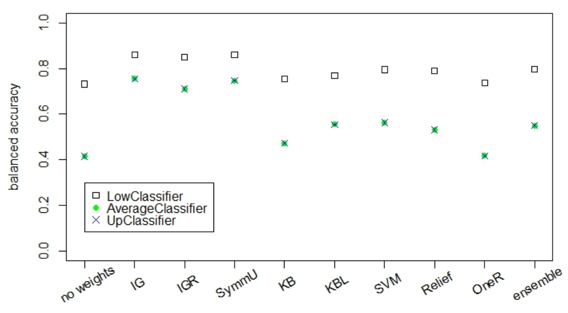

Figure 8. Balanced accuracy, data set: Water2

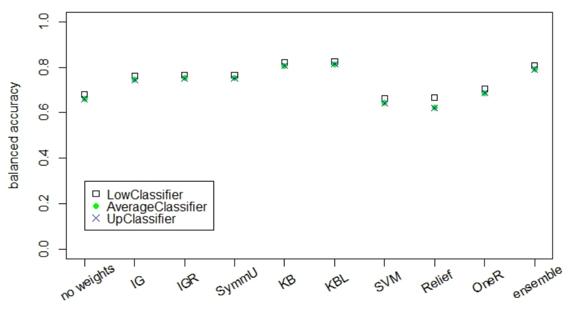

Figure 6. Balanced accuracy, data set: Heart

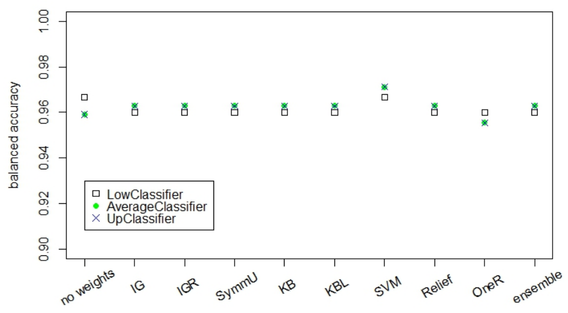

Figure 7. Balanced accuracy, data set: Iris

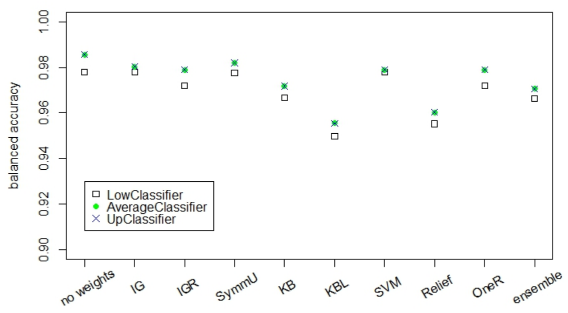

Figure 9. Balanced accuracy, data set: Wine

References

Cornelis, Chris, De Cock, Martine, and Radzikowska, Anna Maria. Vaguely quantified rough sets. In *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, pp. 87–94. Springer, 2007.

Dubois, Didier and Prade, Henri. Rough fuzzy sets and fuzzy rough sets. *International Journal of General System*, 17(2-3):191–209, 1990.

Holte, Robert C. Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1):63–90, 1993.

Jensen, Richard and Cornelis, Chris. Fuzzy-rough nearest neighbour classification and prediction. *Theoretical Computer Science*, 412(42):5871–5884, 2011a.

Jensen, Richard and Cornelis, Chris. Fuzzy-rough nearest neighbour classification. *Transactions on rough sets XIII*, pp. 56–72, 2011b.

Keller, James M, Gray, Michael R, and Givens, James A. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, (4):580–585, 1985.

Kira, Kenji and Rendell, Larry A. The feature selection problem: Traditional methods and a new algorithm. In *Aaai*, volume 2, pp. 129–134, 1992.

Kobayashi, Kingo et al. *Mathematics of information and coding*, volume 203. American Mathematical Soc., 2007.

Kononenko, Igor, Šimec, Edvard, and Robnik-Šikonja, Marko. Overcoming the myopia of inductive learning algorithms with relieff. *Applied Intelligence*, 7(1):39–55, 1997.

Lee, Chang-Hwan, Gutierrez, Fernando, and Dou, Dejing. Calculating feature weights in naive bayes with kullback-leibler measure. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pp. 1146–1151. IEEE, 2011.

Pawlak, Zdzisław. Rough sets. *International Journal of Parallel Programming*, 11(5):341–356, 1982.

Platt, John. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

Radzikowska, Anna Maria and Kerre, Etienne E. A comparative study of fuzzy rough sets. *Fuzzy sets and systems*, 126(2):137–155, 2002.

Shannon, Claude E and Weaver, Warren. The mathematical theory of information. 1949.

Zadeh, Lotfi A. Information and control. *Fuzzy sets*, 8(3): 338–353, 1965.