Hybrid Fuzzy-Rough Rule Induction and Feature Selection

Richard Jensen, Chris Cornelis and Qiang Shen

Abstract— **The automated generation of feature pattern-based if-then rules is essential to the success of many intelligent pattern classifiers, especially when their inference results are expected to be directly human-comprehensible. Fuzzy and rough set theory have been applied with much success to this area as well as to feature selection. Since both applications of rough set theory involve the processing of equivalence classes for their successful operation, it is natural to combine them into a single integrated method that generates concise, meaningful and accurate rules. This paper proposes such an approach, based on fuzzy-rough sets. The algorithm is experimentally evaluated against leading classifiers, including fuzzy and rough rule inducers, and shown to be effective.**

I. Introduction

Fuzzy rule induction forms a major approach to learning robust transparent models. The use of such learning algorithms allows for enhanced transparency in both the learned models themselves and the inferences performed with these models. Many fuzzy rule induction algorithms have been established, mostly for deriving a concise and human-comprehensible set of rules for tasks like classification and prediction. These include, for example, fuzzy association rule mining [4], [29], first-order fuzzy rule generation [8], [22], and linguistic semantics-preserving modeling [20], [24]. However, the efficacy of most of the existing approaches to fuzzy rule induction reduces as the data dimensionality increases. Some methods manage to avoid this, for example standard covering algorithms for rule induction (e.g. RIPPER [5]) that learn rules in an incremental way, with each rule in turn constructed by adding maximally informative features one by one.

A usual technique to address this problem is to employ a feature selection mechanism as a pre-processor. Yet, this adds overheads in the learning process. In addition to this, the feature selection step is typically carried out in isolation to rule induction for filter approaches. This separation can prove costly in that the subset of features returned by the selection process may not be those that are the most useful for the rule induction phase. This has been the motivation behind wrapper approaches to feature selection, but these come with the additional complexity of performing rule induction repeatedly in the search for the optimal set of features. Clearly, a tighter integration of feature selection and rule induction is desirable.

R. Jensen and Q. Shen are with the Department of Computer Science, Aberystwyth University, UK (email: {rkj,qqs}@aber.ac.uk)

C. Cornelis is with the Department of Applied Mathematics and Computer Science, Ghent University, Gent, Belgium (email: Chris.Cornelis@UGent.be)

The work of C. Cornelis was supported by the Research Foundation–Flanders.

Over the past ten years, rough set theory (RST [21]) has become a topic of great interest to researchers and has been applied to many domains. RST offers an alternative approach that preserves the underlying semantics of the data while allowing reasonable generality. It possesses many attributes that are highly desirable; for example, it requires no parameters (eliminating the need for, possibly erroneous, human input) and it finds a minimal knowledge representation. The two main areas of highly successful application for RST are feature selection and rule induction. Since both approaches involve the analysis of equivalence classes generated from the partitioning of the universe of discourse by sets of features, it is natural, to integrate the two, producing a hybrid feature selection/rule induction algorithm that combines the advantages of both. This paper proposes a framework for the development of such techniques. Although the methods presented here are based on a greedy hill-climbing strategy, any search method may be implemented for the purpose, with suitable algorithmic modifications.

The remainder of this paper is structured as follows: in Section II, the necessary theoretical background is provided concerning information and decision systems, and the required fuzzy and rough set concepts. Section III introduces the new hybrid fuzzy-rough rule induction method and provides a simple walkthrough example to illustrate the process. Experimental results that demonstrate the potential of the approach are presented in Section IV. Finally, Section V concludes the paper and outlines some ideas for future work.

II. Preliminaries

A. Existing Work

Due to its recency, there have been very few attempts at developing fuzzy-rough set theory for the purpose of rule induction. Previous work has largely focused on using crisp rough set theory to generate fuzzy rulesets [13], [26] but mainly ignores the direct use of fuzzy-rough concepts.

The induction of gradual decision rules, based on fuzzy-rough hybridization, is given in [9]. For this approach, new definitions of fuzzy lower and upper approximations are constructed that avoid the use of fuzzy logical connectives altogether. Decision rules are induced from lower and upper approximations defined for positive and negative relationships between credibility of premises and conclusions. Only the ordinal properties of fuzzy membership degrees are used. More recently, a fuzzy-rough approach to fuzzy rule induction was presented in [27], where fuzzy reducts are employed to generate rules from data. This method also employs a fuzzy-rough feature selection preprocessing step.

Also of interest is the use of fuzzy-rough concepts in building fuzzy decision trees. Initial research is presented

978-1-4244-3597-5/09/$25.00 ©2009 IEEE

in [1] where a method for fuzzy decision tree construction is given that employs the fuzzy-rough ownership function. This is used to define both an index of fuzzy-roughness and a measure of fuzzy-rough entropy as a node splitting criterion. Traditionally, fuzzy entropy (or its extension) has been used for this purpose. In [16], a fuzzy decision tree algorithm is proposed, based on fuzzy ID3, that incorporates the fuzzy-rough dependency function as a splitting criterion. A fuzzy-rough rule induction method is proposed in [12] for generating certain and possible rulesets from hierarchical data.

B. Information Systems and Fuzzy Indiscernibility

In the context of rough or fuzzy-rough data analysis, an *information system* is a couple $(X, \mathcal{A})$, where $X = \{x_1, \ldots, x_n\}$ and $\mathcal{A} = \{a_1, \ldots, a_m\}$ are finite, non-empty sets of objects and attributes, respectively. Attributes can be either *qualitative* (discrete-valued) or *quantitative* (real-valued). A qualitative attribute a takes values from a finite set, and comparison between values is done on a strict equality basis; this is reflected by the so-called a-indiscernibility relation R_a, defined as $R_a = \{(x, y) \in X^2 | (a(x) = a(y)\}$. When a is quantitative, its values are drawn from a closed interval of real numbers, and compared by means of a fuzzy a-indiscernibility relation R_a, for example,

$$R_a(x, y) = \max\left(\min\left(\frac{a(y) - a(x)}{\sigma_a}, \frac{a(x) - a(y)}{\sigma_a}\right) + 1, 0\right) \quad (1)$$

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

$\forall x, y \in X$, with σ_a denoting the standard deviation of a. Given a subset B of $\mathcal{A}$, the fuzzy B-indiscernibility relation R_B is then defined as, for x and y in X,

$$R_B(x, y) = \mathcal{T}(\underbrace{R_a(x, y)}_{a \in B}) \quad (3)$$

in which $\mathcal{T}$ represents a t-norm. B may contain both quantitative and qualitative attributes; if all attributes are qualitative, R_B is a crisp relation. In general, R_B is a fuzzy tolerance (i.e., reflexive and symmetric) relation. The fuzzy tolerance classes[1] of R_B can be used to approximate fuzzy sets in X (called concepts). Given such a fuzzy set A, its lower and upper approximations w.r.t. R_B are defined by

$$(R_B{\downarrow}A)(y) = \inf_{x \in X} \mathcal{I}(R_B(x, y), A(x)) \quad (4)$$

$$(R_B{\uparrow}A)(y) = \sup_{x \in X} \mathcal{T}(R_B(x, y), A(x)) \quad (5)$$

for all y in X, in which $\mathcal{I}$ represents an implicator[2] and $\mathcal{T}$ is a t-norm. In this paper, the $\min$ t-norm and Kleene-Dienes implicator ($\mathcal{I}(x, y) = \max(1 - x, y)$) are used.

[1]For each $y \in X$, its fuzzy tolerance class $R_B y$ is defined by $R_B y(x) = R_B(x, y), \forall x \in X$.

[2]An implicator is defined as a mapping $\mathcal{I} : [0, 1]^2 \to [0, 1]$ such that $\mathcal{I}$ is decreasing in its first, and increasing in its second component, and satisfies $\mathcal{I}(0, 0) = \mathcal{I}(0, 1) = \mathcal{I}(1, 1) = 1$ and $\mathcal{I}(1, 0) = 0$.

C. Decision Systems and Feature Selection

A *decision system* $(X, \mathcal{A} \cup \{d\})$ is an information system in which d ($d \notin \mathcal{A}$) is a designated attribute called decision. In this paper, it is assumed that decision values are always qualitative; in other words, based on these values, X is partitioned into a number of non-overlapping decision classes X_k ($k = 1, \ldots, p$). Decision systems are often used in the context of classification.

Given $B \subseteq \mathcal{A}$, the fuzzy B-positive region is a fuzzy set in X that contains each object y to the extent that all objects with approximately equal values for the attributes in B, have equal decision values d [7]. In particular,

$$\begin{aligned} POS_B(y) &= \left(\bigcup_{k=1}^{p} R_B{\downarrow}X_k\right)(y) \\ &= \max_{k=1}^{p} \inf_{x \in X} \mathcal{I}(R_B(x, y), A_k(x)) \end{aligned} \quad (6)$$

Given $y \in X_{k^*}$, this formula can be simplified [7] to

$$POS_B(y) = (R_B{\downarrow}X_{k^*})(y) \quad (7)$$

The predictive ability w.r.t. d of the attributes in B is reflected by the ratio γ_B (degree of dependency of d on B), defined as

$$\gamma_B = \frac{|POS_B|}{|POS_{\mathcal{A}}|} = \frac{\sum_{x \in X} POS_B(x)}{\sum_{x \in X} POS_{\mathcal{A}}(x)} \quad (8)$$

A subset B of $\mathcal{A}$ is called a decision superreduct if $\gamma_B = 1$, i.e., B preserves the decision making power of $\mathcal{A}$. If it cannot be further reduced, i.e., there exists no proper subset B' of B such that $POS_{B'} = POS_{\mathcal{A}}$, it is called a decision reduct.

Computing all decision reducts is NP-hard, but in practice it often suffices to generate a single decision superreduct; for this purpose, QUICKREDUCT, a heuristic hill-climbing search algorithm shown in Fig. 1 can be used [16].

```
(1)  B := {}
(2)  do
(3)     T := B
(4)     foreach a ∈ (A \ B)
(5)        if γ_{B∪{a}} > γ_T
(6)           T := B ∪ {a}
(7)     B := T
(8)  until γ_B = γ_A
(9)  return B
```

Fig. 1. The QUICKREDUCT algorithm

D. Crisp Rough Set Rule Induction

In crisp rough set theory, rules can be generated through the use of minimal complexes [10]. Let D be a concept, t an attribute-value pair (a, v), and T be a set of attribute-value pairs. The block of t, denoted $[t]$, is the set of objects for which attribute a has value v. A concept D depends on a set of attribute-value pairs T, if and only if

$$\emptyset \neq \cap\{[t]|t \in T\} \subseteq D \quad (9)$$

T is a minimal complex of D if and only if D depends on T and no proper subset T' of T exists such that D depends on T'.

It is often the case that a minimal complex describes a concept only partially, and hence more than one minimal complex is required to cover a concept. A local covering $\mathbb{T}$ of a concept D is such a collection of minimal complexes, such that the union of all minimal complexes is exactly D and $\mathbb{T}$ is minimal (i.e. contains no spurious attribute-value pairs). The discovery of such local coverings forms the basis of several approaches to rough set rule induction [23]. A partitioning of the universe of discourse by a reduct will always produce equivalence classes that are subsets of the decision concepts and will cover each concept fully. Once a reduct has been found, rules may be extracted from the underlying equivalence classes. In the literature, reducts for the purpose of rule induction are termed global coverings.

The most widely-used approach to rule induction is the LEM2 algorithm [10], which follows a heuristic strategy for creating an initial rule by choosing sequentially the "best" elementary conditions according to some heuristic criteria. Learning examples that match this rule are removed from consideration. The process is repeated iteratively while some learning examples remain uncovered. The resulting set of rules covers all learning examples. In [14], additional factors characterizing rules are taken into account: the strength of matched or partly-matched rules (the total number of cases correctly classified by the rule during training), the number of non-matched conditions, the rule specificity (i.e. length of condition parts). All factors are combined and the strongest decision wins. If no rule is matched, the partly matched rules are considered and the most probable decision is chosen.

III. Hybrid Fuzzy Rule Induction

Feature selection often precedes classification as a preprocessing step, simplifying a decision system by selecting those conditional attributes that are most pertinent to the decision, and eliminating those that are redundant and/or misleading.

As mentioned previously, a common strategy in rough set theory is to induce (fuzzy) rules by overlaying decision reducts over the original (training) decision system and reading off the values. In other words, by partitioning the universe via the features present in a decision reduct, each resulting equivalence class forms a single rule. As the partitioning is produced by a reduct, it is guaranteed that each equivalence class is a subset of, or equal to, a decision concept, meaning that the attribute values that produced this equivalence class are good predictors of the decision concept. The use of a reduct also ensures that each object is covered by the set of rules. A disadvantage of this approach is that the generated rules are often too specific, as each rule antecedent always includes every feature appearing in the final reduct. For this reason, we propose to integrate the rule induction step directly into the feature selection process, generating rules on the fly. In particular, we adapt the QuickReduct algorithm such that, at each step, fuzzy rules that maximally cover the training objects, with a minimal number of attributes, are generated.

For the purposes of combining rule induction and feature selection, rules are constructed from tolerance classes (antecedents) and corresponding decision concepts (consequents). A fuzzy rule, then, is represented as a triple (B, C, D), in which $B \subseteq \mathcal{A}$ is the set of conditional attributes that appear in the rule's antecedent, C is the fuzzy tolerance class of the object that generated the rule and D refers to a decision class (the consequent of the rule). This formulation is used as it provides a fast way of determining rule coverage (the cardinality of C) and rule specificity (the cardinality of B).

A. Algorithm

QuickRules is shown in Fig. 2. It proceeds in a similar way as QuickReduct (indeed, the output B will be identical to the one obtained in Fig. 1), extending this algorithm with the rule induction phase in lines (5)–(7). The rule set is maintained in $Rules$, while the fuzzy set Cov in X records the current degree of coverage of each object in the training data by the current set of rules. Initially, like B, both are equal to the empty set (line 1). The function $covered(Cov)$ returns the set of objects that are maximally covered in Cov, and is defined as

$$covered(Cov) = \{x \in X \mid Cov(x) = POS_{\mathcal{A}}(x)\} \quad (10)$$

This says that an object is considered to be covered by the set of rules if its membership to Cov is equal to that of the positive region of the full feature set. A rule is constructed for an object y and an attribute subset $B \cup \{a\}$ only when it has not been covered maximally yet (line 5), i.e., when $y \notin covered(Cov)$, and it belongs maximally to $POS_{B\cup\{a\}}$ (line 6). This means that a rule is created for y only when y's tolerance class $R_B y$ is fully included in a decision concept, and so the attribute values that generated this tolerance class are good indicators of the concept.

In line (7), the procedure Check is called for the newly created rule; it is added to the rule set only if there are no existing rules with the same or a higher coverage (line (3-4)). If an existing rule has a coverage that is strictly smaller than the new rule's, it is deleted (line (5-6)). Finally, if the new rule meets the criteria for addition, the rule set and coverage are updated. The new coverage is determined by taking the union of the rule's tolerance class with the current coverage. When all objects are fully covered, no further rules are created.

The underlying feature selection process will terminate only when each object belongs to the positive region to the maximal extent, which is also the condition for the rule set to cover all objects maximally. Thus, when the algorithm has finished, the resulting rule set will cover all objects. From the generated set of rules, classification is achieved using Mamdani inference [19]. The complexity of the computation of

(1) $B := \{\}$, $Rules := \{\}$, $Cov := \{\}$
(2) **do**
(3) $T := B$
(4) **foreach** $a \in (\mathcal{A} \setminus B)$
(5) **foreach** $y \in X \setminus covered(Cov)$
(6) **if** $POS_{B \cup \{a\}}(y) = POS_{\mathcal{A}}(y)$
(7) CHECK$(B \cup \{a\}, R_{B \cup \{a\}} y, R_d y)$
(8) **if** $\gamma_{B \cup \{a\}} > \gamma_T$
(9) $T := B \cup \{a\}$
(10) $B := T$
(11) **until** $\gamma_B = \gamma_{\mathcal{A}}$
(12) **return** $B, Rules$

CHECK(B, C, D).
(1) $Add := true$
(2) **foreach** $Rule \in Rules$
(3) **if** $C \subseteq Rule.C$
(4) Add := false; **break**
(5) **elseif** $Rule.C \subset C$
(6) $Rules := Rules \setminus Rule$
(7) **if** Add = true
(8) $Rules := Rules \cup (B, C, D)$
(9) $Cov := Cov \cup C$
(10) **return**

Fig. 2. The QUICKRULES Algorithm

the dependency degree, and the underlying positive regions, is the same as for QUICKREDUCT, $O(|\mathcal{A}| \cdot |X|^2)$.

B. Walkthrough Example

To illustrate the operation of the proposed algorithm, rules are induced via QUICKRULES from the decision system below[3], with 7 objects and 8 conditional attributes, all quantitative:

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	d
x_1	1	101	50	15	36	24.2	0.526	26	0
x_2	8	176	90	34	300	33.7	0.467	58	1
x_3	7	150	66	42	342	34.7	0.718	42	0
x_4	7	187	68	39	304	37.7	0.254	41	1
x_5	0	100	88	60	110	46.8	0.962	31	0
x_6	0	105	64	41	142	41.5	0.173	22	0
x_7	1	95	66	13	38	19.6	0.334	25	0

For this example, the similarity is computed using equation (1), and tolerance class membership represented in vector form.

The algorithm begins (as with QUICKREDUCT) by evaluating individual features. First, a_1 is considered. There are no rules at first, so no objects are covered, hence the first object, x_1, is evaluated. It is found that the degree to which this object belongs to $POS_{\{a_1\}}$ is the same as the degree to which it belongs to $POS_{\mathcal{A}}$, and therefore CHECK$(\{a_1\}, R_{\{a_1\}}x_1, R_d x_1)$ is called, with:

$$\begin{aligned} R_{\{a_1\}}x_1 &= [1.0, 0.0, 0.0, 0.0, 0.73, 0.73, 1.0] \\ R_d x_1 &= [1, 0, 1, 0, 1, 1, 1] \end{aligned}$$

As there are no existing rules, a new rule $(\{a_1\}, R_{\{a_1\}}x_1, R_d x_1)$ is added to the empty rule set and the coverage is updated:

$$Cov = [1.0, 0.0, 0.0, 0.0, 0.73, 0.73, 1.0]$$

It can be seen from this that the rule fully covers the first and last objects, and covers objects x_5 and x_6 to degree 0.73. The algorithm continues to evaluate objects for attribute a_1. When it reaches object x_5 (which is only covered partially by the existing rule set), the algorithm calculates that $POS_{\{a_1\}}(x_5) = POS_{\mathcal{A}}(x_5)$ and so CHECK$(\{a_1\}, R_{\{a_1\}}x_5, R_d x_5)$ is called, with:

$$\begin{aligned} R_{\{a_1\}}x_5 &= [0.73, 0.0, 0.0, 0.0, 1.0, 1.0, 0.73] \\ R_d x_5 &= [1, 0, 1, 0, 1, 1, 1] \end{aligned}$$

As $R_{\{a_1\}}x_5 \not\subseteq R_{\{a_1\}}x_1$ and $R_{\{a_1\}}x_1 \not\subset R_{\{a_1\}}x_5$, a new rule, $(\{a_1\}, R_{\{a_1\}}x_5, R_d x_5)$, is added to the set of rules and the coverage is again updated to become

$$Cov = [1.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0]$$

No further objects are considered for attribute a_1 as the rule set fully covers them. The dependency degree is calculated, producing $\gamma_{\{a_1\}} = 0.61$. Other features are evaluated in the same way, producing no rules:

$$\begin{aligned} \gamma_{\{a_2\}} &= 0.89 & \gamma_{\{a_3\}} &= 0.28 \\ \gamma_{\{a_4\}} &= 0.55 & \gamma_{\{a_5\}} &= 0.70 \\ \gamma_{\{a_6\}} &= 0.56 & \gamma_{\{a_7\}} &= 0.46 \end{aligned}$$

During the evaluation of feature a_8, it is determined that $POS_{\{a_8\}}(x_2) = POS_{\mathcal{A}}(x_2)$, so CHECK$(\{a_8\}, R_{\{a_8\}}x_2, R_d x_2)$ is called, the rule $(\{a_8\}, R_{\{a_8\}}x_2, R_d x_2)$ is added to the rule set and the coverage is updated. The dependency is calculated, resulting in $\gamma_{\{a_8\}} = 0.71$. All features have now been evaluated and the algorithm chooses the feature that causes the greatest increase in dependency degree (line 10); in this case, feature a_2 with a value of 0.89. As this does not equal the dependency degree for the full set of conditional features, the algorithm loops, this time evaluating all combinations of individual features with this feature.

No rules are created for subset $\{a_1, a_2\}$. For subset $\{a_2, a_3\}$, $POS_{\{a_2,a_3\}}(x_3) = POS_{\mathcal{A}}(x_3)$ and $R_{\{a_2,a_3\}}x_3$ is not a subset of existing rules, so the rule $(\{a_2, a_3\}, R_{\{a_2,a_3\}}x_3, R_d x_3)$ is added and the coverage updated. Similarly, the rule $(\{a_2, a_3\}, R_{\{a_2,a_3\}}x_4, R_d x_4)$ is added to the rule set, with Cov updated to become $[1, 1, 1, 1, 1, 1, 1]$. The dependency degree of this subset is

[3]This is a sample taken from the Pima Indians Diabetes dataset located at the UCI Machine Learning repository [2].

$\gamma_{\{a_2,a_3\}} = 1$. The algorithm terminates and outputs the fuzzy rules:

$$(\{a_1\}, R_{\{a_1\}}x_1, R_dx_1),$$
$$(\{a_1\}, R_{\{a_1\}}x_5, R_dx_5),$$
$$(\{a_8\}, R_{\{a_8\}}x_2, R_dx_2),$$
$$(\{a_2, a_3\}, R_{\{a_2,a_3\}}x_3, R_dx_3),$$
$$(\{a_2, a_3\}, R_{\{a_2,a_3\}}x_4, R_dx_4)$$

where, for example, the last rule translates to: if a_2 is $\widetilde{187}$ and a_3 is $\widetilde{68}$ then d is 1. Here, $\widetilde{187}$ and $\widetilde{68}$ are fuzzy numbers whose membership function is defined by the original similarity relation, equation (1).

IV. Experimentation

This section presents the initial experimental evaluation of the proposed method for the task of pattern classification, over 11 benchmark datasets from [2] and [16] with several classifiers. The details of the benchmark datasets used can be found in Table I. The number of conditional features ranges from 8 to 2556 over the datasets, and the number of objects ranges from 120 to 690.

TABLE I
Dataset characteristics

Dataset	n	m	p
Australian	690	14	2
Cleveland	297	13	5
Glass	270	13	7
Heart	214	9	2
Ionosphere	230	34	2
Olitos	120	25	4
Pima	392	8	2
Water 2	390	38	2
Water 3	390	38	3
Web	149	2556	5
Wine	178	13	3

The classifiers themselves are obtained from the WEKA toolkit [28] and ROSE software [23], and are evaluated using their default parameter settings. Additionally, two approaches for nearest neighbor classification are used based on fuzzy sets (FNN) [18] and fuzzy-rough sets (FRNN) [15], as well as a recent method for fuzzy rule induction, QSBA [25]. LEM2 and ModLEM [23], are the leading rough set rule induction methods. The QuickRules algorithm uses the similarity measure given in equation (2). Experimentation is also carried out with several non-rough set-based techniques, namely J48, JRip, PART, SMO and Naive Bayes [28] with default parameters selected. For each algorithm, ten-fold cross validation is performed; the resulting classification accuracies and standard deviations can be seen in Table II for the fuzzy/rough set methods and Table III for the others.

From Table II, it can be seen that QuickRules performs best overall for the fuzzy and rough set-based methods. This demonstrates the power of fuzzy-rough set theory in handling the vagueness and uncertainty often present in data. For the nearest neighbor approaches, the only result of statistical significance where they outperform QuickRules is for the Glass dataset. For seven datasets, their performance is statistically worse than QuickRules. LEM2 performs relatively poorly, particularly for Glass, Ionosphere and Olitos datasets. QSBA performs poorly for the Web dataset in particular, demonstrating the shortcomings of the approach when handling many features. It should also be noted that, unlike QuickRules and QSBA, LEM2 and ModLEM perform a degree of rule pruning during induction which should produce more general rules and better resulting accuracies.

When compared with several leading classifiers (Table III), the proposed technique performs very well. Indeed, the overall performance of QuickRules is slightly worse than that of the support vector method (SMO), and comparable to or better than the remaining classifiers.

V. Conclusions

This paper proposed a novel hybrid approach for fuzzy-rough set rule induction. By performing feature selection and rule induction simultaneously, the generated rulesets are guaranteed to be compact and transparent. The experimental results show that the method performs very well against a range of leading classifiers.

The QuickRules induction algorithm currently does not employ any post-processing procedures to improve rule quality. It is likely that such optimizations will improve classification accuracy further. One possible mechanism for this could be achieved by extending and applying the LEM2 pruning procedure, where antecedents are removed if the underlying equivalence class remains unaffected by their removal. Also, it may be beneficial to employ the VQRS measure [6] instead of the traditional dependency measure in order to better handle noise and uncertainty, particularly when applying the method to real-world problems.

As the induction of rules is based on examining the equivalence classes produced by a partitioning of the universe by a feature subset, any search mechanism formulated in this way can be applied to discover rules. This could be performed (for example) by standard search algorithms such as breadth-first search, best first search, etc or stochastic approaches such as Ant Colony Optimization [3] and Genetic Algorithms [11].

References

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

[2] C. L. Blake and C. J. Merz, UCI Repository of machine learning databases. Irvine, University of California, 1998. http://www.ics.uci.edu/∼mlearn/

[3] E. Bonabeau, M. Dorigo, and G. Theraulez, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press Inc., 1999.

[4] I. Cloete and J. Van Zyl, "Fuzzy rule induction in a set covering framework," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 1, pp. 93–110, 2006.

[5] W.W. Cohen, "Fast effective rule induction," In *Machine Learning: Proceedings of the 12th International Conference*, pp. 115–123, 1995.

[6] C. Cornelis, M. De Cock and A. Radzikowska, "Vaguely Quantified Rough Sets,", Proc. 11th Int. Conf. on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC2007), pp. 87–94, 2007.

[7] C. Cornelis, G. Hurtado Martín, R. Jensen, and D. Ślęzak, "Feature Selection with Fuzzy Decision Reducts", Proc. 3rd Int. Conf. on Rough Sets and Knowledge Technology (RSKT2008), pp. 284–291, 2008.

TABLE II

CLASSIFICATION ACCURACY: FUZZY/ROUGH SET METHODS.

Dataset	QUICKRULES	FRNN	FNN	QSBA	LEM2	ModLEM
AUSTRALIAN	80.28 ± 4.47	70.26 ± 5.18	69.94 ± 4.36	68.84 ± 4.59	86.23 ± 5.24	82.75 ± 3.46
CLEVELAND	56.21 ± 7.54	53.07 ± 6.55	49.75 ± 5.67	51.91 ± 8.50	53.17 ± 5.47	50.15 ± 8.25
GLASS	42.83 ± 12.34	66.54 ± 9.03	68.57 ± 9.62	35.51 ± 8.70	14.98 ± 6.33	62.62 ± 7.96
HEART	80.67 ± 6.83	73.26 ± 8.83	66.11 ± 7.89	83.00 ± 6.07	77.04 ± 10.18	76.3 ± 7.8
IONOSPHERE	91.57 ± 5.62	82.91 ± 7.32	78.00 ± 7.46	82.96 ± 8.71	58.26 ± 12.17	87.39 ± 5.65
OLITOS	72.33 ± 11.48	73.75 ± 11.20	63.25 ± 12.48	78.17 ± 11.77	44.17 ± 9.17	64.17 ± 9.9
PIMA	74.45 ± 4.99	68.94 ± 4.49	73.40 ± 5.02	73.45 ± 5.14	64.32 ± 2.73	73.44 ± 4.84
WATER 2	86.13 ± 4.42	82.33 ± 5.39	77.97 ± 2.66	84.74 ± 4.84	80.26 ± 2.82	86.41 ± 3.98
WATER 3	82.41 ± 4.81	78.18 ± 5.59	74.64 ± 3.77	81.97 ± 5.57	70.51 ± 3.85	84.36 ± 5.67
WEB	63.1 ± 11.89	29.74 ± 4.08	45.55 ± 8.04	1.14 ± 2.70	41.67 ± 11.67	61.05 ± 12.28
WINE	97.75 ± 3.92	93.47 ± 5.18	96.40 ± 4.06	96.17 ± 4.14	73.66 ± 9.25	92.03 ± 8.7

TABLE III

CLASSIFICATION ACCURACY: OTHER CLASSIFIERS.

Dataset	SMO	NaiveBayes	J48	JRip	PART
AUSTRALIAN	84.90 ± 3.63	77.35 ± 4.06	85.61 ± 3.65	85.36 ± 3.31	84.41 ± 3.78
CLEVELAND	58.31 ± 6.15	56.06 ± 6.78	53.39 ± 7.31	54.16 ± 3.64	52.44 ± 7.20
GLASS	57.77 ± 9.10	47.70 ± 9.21	68.08 ± 9.28	67.05 ± 10.69	69.12 ± 8.50
HEART	83.89 ± 6.24	83.59 ± 5.98	78.15 ± 7.42	79.19 ± 6.38	77.33 ± 7.81
IONOSPHERE	82.96 ± 6.93	83.78 ± 7.62	86.13 ± 6.20	87.09 ± 6.92	87.39 ± 6.61
OLITOS	87.92 ± 8.81	78.50 ± 11.31	65.75 ± 12.13	68.83 ± 13.06	67.00 ± 12.86
PIMA	76.80 ± 4.54	75.75 ± 5.32	74.49 ± 5.27	75.18 ± 4.54	73.45 ± 4.51
WATER 2	83.67 ± 4.15	70.28 ± 7.56	83.08 ± 5.45	82.64 ± 5.46	83.79 ± 5.17
WATER 3	86.87 ± 4.36	85.46 ± 4.98	81.59 ± 6.51	82.44 ± 6.63	82.54 ± 5.87
WEB	64.78 ± 10.47	63.41 ± 12.93	57.63 ± 11.31	55.09 ± 12.99	51.50 ± 12.86
WINE	98.70 ± 2.76	97.46 ± 3.86	93.37 ± 5.85	93.18 ± 6.49	92.24 ± 6.22

[8] M. Drobics, U. Bodenhofer, E.P. Klement, "FS-FOIL: an inductive learning method for extracting interpretable fuzzy descriptions," *Internat. J. Approx. Reason.*, vol. 32, pp. 131–152, 2003.

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

[10] J. W. Grzymala-Busse, "Three Strategies to Rule Induction from Data with Numerical Attributes", *Transactions on Rough Sets* 2, pp. 54–62, 2004.

[11] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.

[12] T.P. Hong, Y.L. Liou, and S.L. Wang, "Learning with Hierarchical Quantitative Attributes by Fuzzy Rough Sets," *Proc. Joint Conference on Information Sciences*, Advances in Intelligent Systems Research, 2006.

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

[14] P. Jan, J. W. Grzymala-Busse, S. H. Zdzislaw, "Melanoma prediction using data mining system LERS," *Proceedings of the 25th Annual International Computer Software and Applications Conference*, Chicago, IL, USA, pp. 615-620, 2001.

[15] R. Jensen and C. Cornelis, "A New Approach to Fuzzy-Rough Nearest Neighbour Classification", Proc. 6th Int. Conf. on Rough Sets and Current Trends in Computing (RSCTC 2008), 2008, pp. 310-319.

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

[17] R. Jensen and Q. Shen, "New approaches to Fuzzy-Rough Feature Selection", *IEEE Transactions on Fuzzy Systems*, in press, 2009.

[18] J.M. Keller, M.R. Gray, and J.A. Givens, "A Fuzzy K-Nearest Neighbor Algorithm", *IEEE Transactions on Systems, Man and Cybernetics* 15(4), pp. 580-585, 1985.

[19] E.H. Mamdani, "Applications of fuzzy logic to approximate reasoning using linguistic synthesis," *IEEE Transactions on Computers*, Vol. 26, No. 12, pp. 1182–1191, 1977.

[20] J.G. Marin-Blazquez and Q. Shen, "From approximative to descriptive fuzzy classifiers," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 4, pp. 484–497, 2002.

[21] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*, Kluwer Academic Publishing, 1991.

[22] H. Prade, G. Richard, M. Serrurier, "Enriching relational learning with fuzzy predicates," Proceedings of Principles and Practice of Knowledge Discovery in Databases, pp. 399–410, 2003

[23] B. Predki, Sz. Wilk, "Rough Set Based Data Exploration Using ROSE System," In: Z. W. Ras, A. Skowron, eds., *Foundations of Intelligent Systems*, LNAI 1609, Springer-Verlag, pp. 172–180, 1999.

[24] Z. Qin and J. Lawry, "LFOIL: Linguistic rule induction in the label semantics framework," *Fuzzy Sets and Systems*, vol. 159, no. 4, pp. 435–448, 2008.

[25] K. Rasmani and Q. Shen, "Modifying weighted fuzzy subsethood-based rule models with fuzzy quantifiers," Proceedings of the IEEE International Conference on Fuzzy Systems, pp. 1687–1694, 2004.

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

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

[28] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools with Java implementations*, Morgan Kaufmann Publishers, San Francisco, 2000.

[29] D. Xie, "Fuzzy associated rules discovered on effective reduced database algorithm," Proceedings of the IEEE International Conference on Fuzzy Systems, pp. 779–784, 2005.