Semantics-Preserving Dimensionality Reduction: Rough and Fuzzy-Rough-Based Approaches

Jensen, Richard; Shen, Qiang

Published in:
IEEE Transactions on Knowledge and Data Engineering

DOI:
10.1109/TKDE.2004.96

Publication date:
2004

Citation for published version (APA):
Jensen, R., & Shen, Q. (2004). Semantics-Preserving Dimensionality Reduction: Rough and Fuzzy-Rough-Based Approaches. *IEEE Transactions on Knowledge and Data Engineering*, *16*(12), 1457-1471. https://doi.org/10.1109/TKDE.2004.96

Semantics-Preserving Dimensionality Reduction: Rough and Fuzzy-Rough-Based Approaches

Richard Jensen and Qiang Shen

Abstract—Semantics-preserving dimensionality reduction refers to the problem of selecting those input features that are most predictive of a given outcome; a problem encountered in many areas such as machine learning, pattern recognition, and signal processing. This has found successful application in tasks that involve data sets containing huge numbers of features (in the order of tens of thousands), which would be impossible to process further. Recent examples include text processing and Web content classification. One of the many successful applications of rough set theory has been to this feature selection area. This paper reviews those techniques that preserve the underlying semantics of the data, using crisp and fuzzy rough set-based methodologies. Several approaches to feature selection based on rough set theory are experimentally compared. Additionally, a new area in feature selection, feature grouping, is highlighted and a rough set-based feature grouping technique is detailed.

Index Terms—Dimensionality reduction, feature selection, feature transformation, rough selection, fuzzy-rough selection.

1 Introduction

Many problems in machine learning involve high-dimensional descriptions of input features. It is therefore not surprising that much research has been carried out on dimensionality reduction [12], [26], [29], [30], [31]. However, existing work tends to destroy the underlying semantics of the features after reduction (e.g. transformation-based approaches [13]) or require additional information about the given data set for thresholding (e.g. entropy-based approaches [32]). A technique that can reduce dimensionality using information contained within the data set and that preserves the meaning of the features (i.e., semantics-preserving) is clearly desirable. Rough set theory (RST) can be used as such a tool to discover data dependencies and to reduce the number of attributes contained in a data set using the data alone and no additional information [38], [41].

Over the past 10 years, RST has indeed become a topic of great interest to researchers and has been applied to many domains. Given a data set with discretized attribute values, it is possible to find a subset (termed a *reduct*) of the original attributes using RST that are the most informative; all other attributes can be removed from the data set with minimal information loss. From the dimensionality reduction perspective, informative features are those that are most useful in determining classifications from their values.

- *R. Jensen is with the Centre for Intelligent Systems and Their Applications, School of Informatics, The University of Edinburgh, Room 3.14, Appleton Tower, Crichton Street, Edinburgh, EG8 9LE. E-mail: richjens@dai.ed.ac.uk.*
- *Q. Shen is with the Department of Computer Science, University of Wales, Aberystwyth, Ceredigion, SY23 3DB, Wales, UK. E-mail: qqs@aber.ac.uk.*

Manuscript received 23 June 2003; revised 11 Feb. 2004; accepted 27 Apr. 2004.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0102-0603.

However, it is most often the case that the values of attributes may be both crisp and *real-valued*, and this is where traditional rough set theory encounters a problem. It is not possible in the original theory to say whether two attribute values are similar and to what extent they are the same; for example, two close values may only differ as a result of noise, but in RST, they are considered to be as different as two values of a different order of magnitude. As a result of this, extensions to the original theory have been proposed, for example, those based on similarity or tolerance relations [55], [61], [62].

It is, therefore, desirable to develop techniques to provide the means of data reduction for crisp and real-value attributed data sets which utilizes the extent to which values are similar. This can be achieved through the use of *fuzzy-rough* sets. Fuzzy-rough sets encapsulate the related but distinct concepts of vagueness (for fuzzy sets [70]) and indiscernibility (for rough sets), both of which occur as a result of uncertainty in knowledge [17].

This review focuses on those recent techniques for feature selection that employ a rough-set-based methodology for this purpose, highlighting current trends and future directions for this promising area. The second section introduces rough set fundamentals and extensions which enable various approaches to feature selection. Several of these are evaluated experimentally and compared. Section 3 introduces the fuzzy extension to rough sets, fuzzy-rough sets, and details how this may be applied to the feature selection problem, with the aid of a simple example data set. Rough set-based feature grouping is also discussed. The review is concluded in Section 4.

2 Rough Selection

Rough set theory [18], [37], [48], [56], [57] is an extension of conventional set theory that supports approximations in

TABLE 1
An Example Data Set

$x \in \mathbb{U}$	a	b	c	d	$\Rightarrow$	e
0	1	0	2	2		0
1	0	1	1	1		2
2	2	0	0	1		1
3	1	1	0	2		2
4	1	0	2	0		1
5	2	2	0	1		1
6	2	1	1	1		2
7	0	1	1	0		1

decision making. It possesses many features in common (to a certain extent) with the Dempster-Shafer theory of evidence [54] and fuzzy set theory [39], [68]. The rough set itself is the approximation of a vague concept (set) by a pair of precise concepts, called lower and upper approximations, which are a classification of the domain of interest into disjoint categories. The lower approximation is a description of the domain objects which are known with certainty to belong to the subset of interest, whereas the upper approximation is a description of the objects which possibly belong to the subset. This section focuses on several rough set-based techniques for feature selection. Some of the techniques described here can be found in rough set systems available online [44], [45].

To illustrate the operation of these, an example data set (Table 1) will be used. Here, the table consists of four conditional features (a, b, c, d), one decision feature (e) and eight objects. The task of feature selection here is to choose the smallest subset of these conditional features so that the resulting reduced data set remains consistent with respect to the decision feature. A data set is consistent if for every set of objects whose attribute values are the same, the corresponding decision attributes are identical. Throughout this section, the terms attribute and feature are used interchangeably.

2.1 Rough Set Attribute Reduction

Rough Set Attribute Reduction (RSAR) [10], [22], [38], [53], [63] provides a filter-based tool by which knowledge may be extracted from a domain in a concise way; retaining the information content while reducing the amount of knowledge involved. The main advantage that rough set analysis has is that it requires no additional parameters to operate other than the supplied data [19]. It works by making use of the granularity structure of the data only. This is a major difference when compared with Dempster-Shafer theory [49] and fuzzy set theory which require probability assignments and membership values, respectively. However, this does not mean that *no* model assumptions are made. In fact, by using only the given information, the theory assumes that the data is a true and accurate reflection of the real world (which may not be the case). The numerical and other contextual aspects of the data are ignored which may seem to be a significant omission, but it keeps model assumptions to a minimum.

2.1.1 Theoretical Background

Central to RSAR is the concept of indiscernibility. Let $I = (\mathbb{U}, \mathbb{A})$ be an information system, where $\mathbb{U}$ is a nonempty set of finite objects (the universe) and $\mathbb{A}$ is a nonempty finite set of attributes such that $a : \mathbb{U} \rightarrow V_a$ for every $a \in \mathbb{A}$. With any $P \subseteq \mathbb{A}$ there is an associated equivalence relation $IND(P)$:

$$IND(P) = \{(x, y) \in \mathbb{U}^2 \mid \forall a \in P, a(x) = a(y)\}.$$

The partition of $\mathbb{U}$, generated by *IND(P)* is denoted as $\mathbb{U}$ / *IND(P)* and can be calculated as follows:

$$\mathbb{U}/IND(P) = \otimes\{a \in P : \mathbb{U}/IND(\{a\})\}, \quad (1)$$

where

$$A \otimes B = \{X \cap Y : \forall X \in A, \forall Y \in B, X \cap Y \neq \emptyset\}. \quad (2)$$

If $(x, y) \in IND(P)$, then x and y are indiscernible by attributes from P. The equivalence classes of the P-indiscernibility relation are denoted $[x]_P$. For the illustrative example, if $P = \{b, c\}$, then objects 1, 6, and 7 are indiscernible; as are objects 0 and 4. *IND(P)* creates the following partition of $\mathbb{U}$:

$$\begin{aligned}\mathbb{U}/IND(P) &= \mathbb{U}/IND(b) \otimes \mathbb{U}/IND(c)\\ &= \{\{0,2,4\},\{1,3,6,7\},\{5\}\} \otimes \{\{2,3,5\},\{1,6,7\},\{0,4\}\}\\ &= \{\{2\},\{0,4\},\{3\},\{1,6,7\},\{5\}\}.\end{aligned}$$

Let $X \subseteq \mathbb{U}$. X can be approximated using only the information contained within P by constructing the P-*lower* and P-*upper* approximations of X:

$$\underline{P}X = \{x \mid [x]_P \subseteq X\}, \quad (3)$$

$$\overline{P} = \{x \mid [x]_P \cap X \neq \emptyset\}. \quad (4)$$

Let P and Q be equivalence relations over $\mathbb{U}$, then the positive, negative, and boundary regions can be defined as:

$$\begin{aligned}POS_P(Q) &= \bigcup_{X \in \mathbb{U}/Q} \underline{P}X\\ NEG_P(Q) &= \mathbb{U} - \bigcup_{X \in \mathbb{U}/Q} \overline{P}X\\ BND_P(Q) &= \bigcup_{X \in \mathbb{U}/Q} \overline{P}X - \bigcup_{X \in \mathbb{U}/Q} \underline{P}X.\end{aligned}$$

The positive region contains all objects of $\mathbb{U}$ that can be classified to classes of $\mathbb{U}/Q$ using the knowledge in attributes P. The boundary region, $BND_P(Q)$, is the set of objects that can possibly, but not certainly, be classified in this way. The negative region, $NEG_P(Q)$, is the set of objects that cannot be classified to classes of $\mathbb{U}/Q$. For example, let $P = \{b, c\}$ and $Q = \{e\}$, then

$$\begin{aligned}POS_{IND(P)}(Q) &= \bigcup\{\emptyset,\{2,5\},\{3\}\} = \{2,3,5\},\\ NEG_{IND(P)}(Q) &= \mathbb{U} - \bigcup\{\{0,4\},\{2,0,4,1,6,7,5\}\\ &\qquad \{3,1,6,7\}\} = \emptyset,\\ BND_{IND(P)}(Q) &= \mathbb{U} - \{2,3,5\} = \{0,1,4,6,7\}.\end{aligned}$$

This means that objects 2, 3, and 5 can certainly be classified as belonging to a class in attribute e, when considering attributes b and c. The rest of the objects cannot be classified as the information that would make them discernible is absent.

An important issue in data analysis is discovering dependencies between attributes. Intuitively, a set of attributes Q depends totally on a set of attributes P, denoted $P \Rightarrow Q$, if all attribute values from Q are uniquely determined by values of attributes from P. If there exists a functional dependency between values of Q and P, then Q depends totally on P. Dependency can be defined in the following way:

For $P, Q \subset \mathbb{A}$, it is said that Q depends on P in a degree $k (0 \leq k \leq 1)$, denoted $P \Rightarrow_k Q$, if

$$k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|\mathbb{U}|}. \tag{5}$$

If $k = 1$, Q depends totally on P, if $0 < k < 1$, Q depends partially (in a degree k) on P, and if $k = 0$, then Q does not depend on P. In the example, the degree of dependency of attribute $\{e\}$ from the attributes $\{b, c\}$ is:

$$\gamma_{\{b,c\}}(\{e\}) = \frac{|POS_{\{b,c\}}(\{e\})|}{|\mathbb{U}|} = \frac{|\{2,3,5\}|}{|\{0,1,2,3,4,5,6,7\}|} = \frac{3}{8}.$$

By calculating the change in dependency when an attribute is removed from the set of considered conditional attributes, a measure of the significance of the attribute can be obtained. The higher the change in dependency, the more significant the attribute is. If the significance is 0, then the attribute is dispensable. More formally, given P, Q and an attribute $a \in P$,

$$\sigma_P(Q, a) = \gamma_P(Q) - \gamma_{P-\{a\}}(Q). \tag{6}$$

For example, if$P = \{a, b, c\}$ and $Q = e$, then

$$\begin{aligned}
\gamma_{\{a,b,c\}}(\{e\}) &= |\{2,3,5,6\}|/8 = 4/8 \\
\gamma_{\{a,b\}}(\{e\}) &= |\{2,3,5,6\}|/8 = 4/8 \\
\gamma_{\{b,c\}}(\{e\}) &= |\{2,3,5\}|/8 = 3/8 \\
\gamma_{\{a,c\}}(\{e\}) &= |\{2,3,5,6\}|/8 = 4/8.
\end{aligned}$$

And, calculating the significance of the three attributes gives:

$$\begin{aligned}
\sigma_P(Q, a) &= \gamma_{\{a,b,c\}}(\{e\}) - \gamma_{\{b,c\}}(\{e\}) = 1/8 \\
\sigma_P(Q, b) &= \gamma_{\{a,b,c\}}(\{e\}) - \gamma_{\{a,c\}}(\{e\}) = 0 \\
\sigma_P(Q, c) &= \gamma_{\{a,b,c\}}(\{e\}) - \gamma_{\{a,b\}}(\{e\}) = 0.
\end{aligned}$$

From this, it follows that attribute a is indispensable, but attributes b and c can be dispensed with.

2.1.2 Reduction Method

The reduction of attributes is achieved by comparing equivalence relations generated by sets of attributes. Attributes are removed so that the reduced set provides the same quality of classification as the original. A *reduct* is defined as a subset of minimal cardinality R_{min} of the conditional attribute set $\mathbb{C}$ such that $\gamma_R(\mathbb{D}) = \gamma_{\mathbb{C}}(\mathbb{D})$.

$$\mathrm{R} = \{X : X \subseteq \mathbb{C}, \gamma_X(\mathbb{D}) = \gamma_{\mathbb{C}}(\mathbb{D})\}, \tag{7}$$

$$\mathrm{R}_{\min} = \{X : X \in \mathrm{R}, \forall Y \in \mathrm{R}, |X| \leq |Y|\}. \tag{8}$$

The intersection of all the sets in $\mathrm{R}_{\min}$ is called the *core*, the elements of which are those attributes that cannot be eliminated without introducing more contradictions to the data set. In RSAR, a subset with minimum cardinality is searched for.

Using the example, the dependencies for all possible subsets of $\mathbb{C}$ can be calculated as:

$$\begin{array}{ll}
\gamma_{\{a,b,c,d\}}(\{e\}) = 8/8 & \gamma_{\{b,c\}}(\{e\}) = 3/8 \\
\gamma_{\{a,b,c\}}(\{e\}) = 4/8 & \gamma_{\{b,d\}}(\{e\}) = 8/8 \\
\gamma_{\{a,b,d\}}(\{e\}) = 8/8 & \gamma_{\{c,d\}}(\{e\}) = 8/8 \\
\gamma_{\{a,c,d\}}(\{e\}) = 8/8 & \gamma_{\{a\}}(\{e\}) = 0/8 \\
\gamma_{\{b,c,d\}}(\{e\}) = 8/8 & \gamma_{\{b\}}(\{e\}) = 1/8 \\
\gamma_{\{a,b\}}(\{e\}) = 4/8 & \gamma_{\{c\}}(\{e\}) = 0/8 \\
\gamma_{\{a,c\}}(\{e\}) = 4/8 & \gamma_{\{d\}}(\{e\}) = 2/8 \\
\gamma_{\{a,d\}}(\{e\}) = 3/8. &
\end{array}$$

Note that the given data set is consistent since $\gamma_{\{a,b,c,d\}}(\{e\}) = 1$. The minimal reduct set for this example is:

$$\mathrm{R}_{\min} = \{\{b, d\}, \{c, d\}\}.$$

If $\{b, d\}$ is chosen, then the data set can be reduced as in Table 2. Clearly, each object can be uniquely classified according to the attribute values remaining.

TABLE 2
Reduced Data Set

$x \in \mathbb{U}$	b	d	$\Rightarrow$	e
0	0	2		0
1	1	1		2
2	0	1		1
3	1	2		2
4	0	0		1
5	2	1		1
6	1	1		2
7	1	0		1

The problem of finding a reduct of an information or decision system has been the subject of much research [2], [22], [58], [53]. The most basic solution to locating such a subset is to simply generate *all* possible subsets and retrieve those with a maximum rough set dependency degree. Obviously, this is an expensive solution to the problem and is only practical for very simple data sets. Most of the time only one reduct is required, so all the calculations involved in discovering the rest are pointless.

To improve the performance of the above method, an element of pruning can be introduced. By noting the cardinality of any prediscovered reducts, the current possible subset can be ignored if it contains more elements. However, a better approach is needed—one that will avoid wasted computational effort.

QUICKREDUCT($\mathbb{C}$,$\mathbb{D}$).
$\mathbb{C}$, the set of all conditional features;
$\mathbb{D}$, the set of decision features.

(1) $R \leftarrow \{\}$
(2) **do**
(3) $T \leftarrow R$
(4) $\forall x \in (\mathbb{C} - R)$
(5) **if** $\gamma_{R\cup\{x\}}(\mathbb{D}) > \gamma_T(D)$
(6) $T \leftarrow R \cup \{x\}$
(7) $R \leftarrow T$
(8) **until** $\gamma_R(\mathbb{D}) == \gamma_{\mathbb{C}}(\mathbb{D})$
(9) **return** R

Fig. 1. The QUICKREDUCT Algorithm.

The QUICKREDUCT Algorithm given in Fig. 1 (adapted from [10]), attempts to calculate a reduct without exhaustively generating all possible subsets. It starts off with an empty set and adds, in turn, one at a time, those attributes that result in the greatest increase in the rough set dependency metric, until this produces its maximum possible value for the data set. Other such techniques may be found in [40].

According to the QUICKREDUCT algorithm, the dependency of each attribute is calculated, and the best candidate chosen. In Fig. 2, this stage is illustrated using the example data set. As attribute d generates the highest dependency degree, then that attribute is chosen and the sets $\{a, d\}$, $\{b, d\}$, and $\{c, d\}$ are evaluated. This process continues until the dependency of the reduct equals the consistency of the data set (1 if the data set is consistent). The generated reduct shows the way of reducing the dimensionality of the original data set by eliminating those conditional attributes that do not appear in the set.

This, however, is not guaranteed to find a *minimal* subset as has been shown in [11]. Using the dependency function to discriminate between candidates may lead the search down a nonminimal path. It is impossible to predict which combinations of attributes will lead to an optimal reduct based on changes in dependency with the addition or deletion of single attributes. It does result in a close-to-minimal subset, though, which is still useful in greatly reducing data set dimensionality.

In [11], a potential solution to this problem has been proposed whereby the QUICKREDUCT algorithm is altered, making it into an n-lookahead approach. However, even this cannot guarantee a reduct unless n is equal to the original number of attributes, but this reverts back to generate-and-test. It still suffers from the same problem as the original QUICKREDUCT, i.e., it is impossible to tell at any stage whether the current path will be the shortest to a reduct.

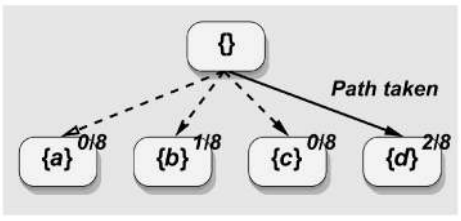

Fig. 2. Branches of the search space.

TABLE 3
The Decision-Relative Discernibility Matrix

$x \in \mathbb{U}$	0	1	2	3	4	5	6	7
0								
1	a, b, c, d							
2	a, c, d	a, b, c						
3	b, c		a, b, d					
4	d	a, b, c, d		b, c, d				
5	a, b, c, d	a, b, c		a, b, d				
6	a, b, c, d		b, c		a, b, c, d	b, c		
7	a, b, c, d	d		a, c, d			a, d	

It is interesting to note that the rough set degree of dependency measure is very similar to the consistency criterion used by the FOCUS algorithm and others [1], [46]. In FOCUS, a breadth-first search is employed such that any subset is rejected if this produces at least one inconsistency. If this is converted into a guided search using the consistency measure as a heuristic, it should behave exactly as QUICKREDUCT. Consistency is defined as the number of discernible objects out of the entire object set—exactly that of the dependency measure.

2.2 Discernibility Matrix Approach

Many applications of rough sets to feature selection make use of discernibility matrices for finding reducts. A discernibility matrix [27], [53] of a decision table $D = (\mathbb{U}, \mathbb{C} \cup \mathbb{D})$ is a symmetric $|\mathbb{U}| \times |\mathbb{U}|$ matrix with entries defined as:

$$d_{ij} = \{a \in \mathbb{C} | a(x_i) \neq a(x_j)\} \quad i, j = 1, \ldots, |\mathbb{U}|. \tag{9}$$

Each d_{ij} contains those attributes that differ between objects i and j. For finding reducts, the decision-relative discernibility matrix is of more interest. This only considers those object discernibilities that occur when the corresponding decision attributes differ. Returning to the example data set, the decision-relative discernibility matrix found in Table 3 is produced. For example, it can be seen from the table that objects 0 and 1 differ in each attribute. Although some attributes in objects 1 and 3 differ, their corresponding decisions are the same so no entry appears in the decision-relative matrix. Grouping all entries containing single attributes forms the core of the data set (those attributes appearing in *every* reduct). Here, the core of the data set is $\{d\}$.

From this, the discernibility function can be defined. This is a concise notation of how each object within the data set may be distinguished from the others. A discernibility function f_D is a Boolean function of m Boolean variables $a_1^*, \ldots, a_m^*$ (corresponding to the attributes $a_1, \ldots, a_m$) defined as below:

$$f_D(a_1^*, \ldots, a_m^*) = \wedge\{\vee c_{ij}^* | 1 \leq j \leq i \leq |\mathbb{U}|, c_{ij} \neq \emptyset\}, \tag{10}$$

where $c_{ij}^* = \{a^* | a \in c_{ij}\}$. By finding the set of all prime implicants of the discernibility function, all the minimal reducts of a system may be determined. From Table 3, the

decision-relative discernibility function is (with duplicates removed):

$$f_D(a,b,c,d) = \{a \vee b \vee c \vee d\} \wedge \{a \vee c \vee d\} \wedge \{b \vee c\} \wedge \{d\} \wedge \{a \vee b \vee c\} \wedge \{a \vee b \vee d\} \wedge \{b \vee c \vee d\} \wedge \{a \vee d\}.$$

Further simplification can be performed by removing those sets that are supersets of others:

$$f_D(a,b,c,d) = \{b \vee c\} \wedge \{d\}$$

The reducts of the data set may be obtained by converting the above expression from conjunctive normal form to disjunctive normal form (without negations). Hence, the minimal reducts are $\{b,d\}$ and $\{c,d\}$. Although this is guaranteed to discover all minimal subsets, it is a costly operation rendering the method impractical for even medium-sized data sets.

For certain applications, a single minimal subset is all that is required for data reduction. For example, dimensionality reduction within text classification tends to use only one subset to remove unnecessary keywords [25], [65]. This has led to approaches that consider finding individual shortest prime implicants from the discernibility function. A common method is to incrementally add those attributes that occur with the most frequency in the function, removing any clauses containing the attributes, until all clauses are eliminated [35], [66]. However, even this does not ensure that a minimal subset is found—the search can proceed down nonminimal paths.

It may also be desirable to locate several minimal subsets for some applications [24], [67]. Once a collection of such subsets has been found, a choice is made as to which of these are the most informative for the application task at hand. This decision can be made manually, or by the use of a suitable measure such as entropy [42] to distinguish between the subsets.

2.3 Reduction with Variable Precision Rough Sets

Variable precision rough sets (VPRS) [72] attempts to improve upon rough set theory by relaxing the subset operator. It was proposed to analyze and identify data patterns which represent statistical trends rather than functional. The main idea of VPRS is to allow objects to be classified with an error smaller than a certain predefined level. Let $X, Y \subseteq \mathbb{U}$, the relative classification error is defined by:

$$c(X,Y) = 1 - \frac{|X \cap Y|}{|X|}.$$

Observe that $c(X,Y) = 0$ if and only if $X \subseteq Y$. A degree of inclusion can be achieved by allowing a certain level of error, β, in classification:

$$X \subseteq_\beta Y \text{ iff } c(X,Y) \leq \beta, \quad 0 \leq \beta < 0.5.$$

Using $\subseteq_\beta$ instead of $\subseteq$, the β-upper and β-lower approximations of a set X can be defined as:

$$\underline{R}_\beta X = \bigcup \{[x]_R \in \mathbb{U}/R | [x]_R \subseteq_\beta X\}$$
$$\overline{R}_\beta X = \bigcup \{[x]_R \in \mathbb{U}/R | c([x]_R, X) < 1 - \beta\}.$$

Note that $\underline{R}_\beta X = \underline{R} X$ for $\beta = 0$. The positive, negative, and boundary regions in the original rough set theory can now be extended to:

$$POS_{R,\beta}(X) = \underline{R}_\beta X, \tag{11}$$

$$NEG_{R,\beta}(X) = \mathbb{U} - \overline{R}_\beta X, \tag{12}$$

$$BND_{R,\beta}(X) = \overline{R}_\beta X - \underline{R}_\beta X. \tag{13}$$

Returning to the example data set in Table 1, (11) can be used to calculate the β-positive region for $R = \{b,c\}$, $X = \{e\}$, and $\beta = 0.4$. Setting β to this value means that a set is considered to be a subset of another if they share about half the number of elements. The partitions of the universe of objects for R and X are:

$$\mathbb{U}/R = \{\{2\},\{0,4\},\{3\},\{1,6,7\},\{5\}\}$$
$$\mathbb{U}/X = \{\{0\},\{1,3,6\},\{2,4,5,7\}\}.$$

For each set $A \in \mathbb{U}/R$ and $B \in \mathbb{U}/X$, the value of $c(A,B)$ must be less than β if the equivalence class A is to be included in the β-positive region. Considering $A = \{2\}$ gives

$$\mathrm{c}(\{2\},\{0\}) = 1 > \beta$$
$$\mathrm{c}(\{2\},\{1,3,6\}) = 1 > \beta$$
$$\mathrm{c}(\{2\},\{2,4,5,7\}) = 0 < \beta.$$

So, object 2 is added to the β-positive region as it is a β-subset of $\{2,4,5,7\}$ (and is, in fact, a traditional subset of the equivalence class). Taking $A = \{1,6,7\}$, a more interesting case is encountered:

$$\mathrm{c}(\{1,6,7\},\{0\}) = 1 > \beta$$
$$\mathrm{c}(\{1,6,7\},\{1,3,6\}) = 0.3333 < \beta$$
$$\mathrm{c}(\{1,6,7\},\{2,4,5,7\}) = 0.6667 > \beta.$$

Here, the objects 1, 6, and 7 are included in the β-positive region as the set $\{1,6,7\}$ is a β-subset of $\{1,3,6\}$. Calculating the subsets in this way leads to the following β-positive region:

$$POS_{R,\beta}(X) = \{1,2,3,5,6,7\}.$$

Compare this with the positive region generated previously: $\{2,3,5\}$. Objects 1, 6, and 7 are now included due to the relaxation of the subset operator. Consider a decision table $A = (\mathbb{U}, \mathbb{C} \cup \mathbb{D})$, where $\mathbb{C}$ is the set of conditional attributes and $\mathbb{D}$ is the set of decision attributes. The β-positive region of an equivalence relation Q on $\mathbb{U}$ may be determined by

$$POS_{R,\beta}(Q) = \bigcup_{X \in \mathbb{U}/Q} \underline{R}_\beta X,$$

where R is also an equivalence relation on $\mathbb{U}$. This can then be used to calculate dependencies and, thus, determine β-reducts. The dependency function becomes:

$$\gamma_{R,\beta}(Q) = \frac{|POS_{R,\beta}(Q)|}{|\mathbb{U}|}.$$

It can be seen that the QUICKREDUCT algorithm outlined previously can be adapted to incorporate the reduction method built upon VPRS theory. By supplying a suitable β value to the algorithm, the β-lower approximation, β-positive region, and β-dependency can replace the traditional calculations. This will result in a more approximate final reduct, which may be a better generalization when encountering unseen data. Additionally, setting β to 0 forces such a method to behave exactly like RSAR.

Extended classification of reducts in the VPRS approach may be found in [6], [7], [28]. However, the variable precision approach requires the additional parameter β which has to be specified from the start. By repeated experimentation, this parameter can be suitably approximated. However, problems arise when searching for true reducts as VPRS incorporates an element of imprecision in determining the number of classifiable objects.

2.4 Dynamic Reducts

Reducts generated from information systems are sensitive to changes in the system. This can be seen by removing a randomly chosen set of objects from the original object set. Those reducts frequently occurring in random subtables can be considered to be stable; it is these reducts that are encompassed by *dynamic reducts* [3]. Let $\mathcal{A} = (\mathbb{U}, \mathbb{C} \cup d)$ be a decision table, then any system $\mathcal{B} = (\mathbb{U}', \mathbb{C} \cup d)(\mathbb{U}' \subseteq \mathbb{U})$ is called a subtable of $\mathcal{A}$. If $\mathcal{F}$ is a family of subtables of $\mathcal{A}$, then

$$DR(\mathcal{A}, \mathcal{F}) = Red(\mathcal{A}, d) \cap \left\{ \bigcap_{\mathcal{B} \in \mathcal{F}} Red(\mathcal{B}, d) \right\}$$

defines the set of $\mathcal{F}$-dynamic reducts of $\mathcal{A}$. From this definition, it follows that a relative reduct of $\mathcal{A}$ is dynamic if it is also a reduct of all subtables in $\mathcal{F}$. In most cases, this is too restrictive, so a more general notion of dynamic reducts is required.

By introducing a threshold, $0 \leq \epsilon \leq 1$, the concept of $(\mathcal{F}, \epsilon)$-dynamic reducts can here be defined:

$$DR_\epsilon(\mathcal{A}, \mathcal{F}) = \{C \in Red(\mathcal{A}, d) : s_F(C) \geq \epsilon\},$$

where

$$s_F(C) = \frac{|\{\mathcal{B} \in \mathcal{F} : C \in Red(\mathcal{B}, d)\}|}{|\mathcal{F}|}$$

is the $\mathcal{F}$-stability coefficient of C. This lessens the previous restriction that a dynamic reduct must appear in *every* generated subtable. Now, a reduct is considered to be dynamic if it appears in a certain percentage of subtables, determined by the value ϵ. For example, by setting ϵ to 0.5, a reduct is considered to be dynamic if it appears in at least half of the subtables. Note that if $\mathcal{F} = \{\mathcal{A}\}$, then $DR(\mathcal{A}, \mathcal{F}) = Red(\mathcal{A}, d)$. Dynamic reducts may then be calculated according to the algorithm given in Fig. 3. First, all reducts are calculated for the given information system, $\mathcal{A}$. Then, the new subsystems $\mathcal{A}_j$ are generated by randomly deleting one or more rows from $\mathcal{A}$. All reducts are found for each subsystem, and the dynamic reducts are computed using $s_F(C, R)$ which denotes the significance factor of reduct C within all reducts found, R.

DynamicRed($\mathcal{A}$,ϵ,its).
$\mathcal{A}$, the original decision table;
ϵ, the dynamic reduct threshold;
its, the number of iterations.

```
(1)  R ← {}
(2)  A ← calculateAllReducts(A)
(3)  for j=1...its
(4)    A_j ← deleteRandomRows(A)
(5)    R ← R∪ calculateAllReducts(A_j)
(6)  ∀C ∈ A
(7)    if s_F(C, R) ≥ ε
(8)      output C
```

Fig. 3. Dynamic Reduct algorithm.

Returning to the example decision table (call this $\mathcal{A}$), the first step is to calculate all its reducts. This produces the set of all reducts $A = \{\{b, d\}, \{c, d\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}\}$. The reduct $\{a, b, c, d\}$ is not included as this will always be a reduct of any generated subtable (it is the full set of conditional attributes). The next step randomly deletes a number of rows from the original table $\mathcal{A}$. From this, all reducts are again calculated producing, for one subtable this might be $R = \{\{b, d\}, \{b, c, d\}, \{a, b, d\}\}$. In this case, the subset $\{c, d\}$ is not a reduct (though it was for the original data set). If the number of iterations is set to just one, and if ϵ is set to a value less than 0.5 (implying that a reduct should appear in half of the total number of discovered reducts), then the reduct $\{c, d\}$ is deemed not to be a dynamic reduct.

Intuitively, this is based on the hope that by finding stable reducts they will be more representative of the real world, i.e., it is more likely that they will be reducts for unseen data. A comparison of dynamic and nondynamic approaches can be found in [4], where various methods were tested on extracting laws from decision tables. In the experiments, the dynamic method and the conventional RS method both performed well. In fact, it appears that the RS method has, on average, a lower error rate of classification than the dynamic RS method.

A disadvantage of this dynamic approach is that several subjective choices have to be made before the dynamic reducts can be found (for instance, the choice of the value of ϵ); these values are not contained in the data. Also, the huge complexity of finding all reducts within subtables forces the use of heuristic techniques such as genetic algorithms. For large data sets, this step may well be too costly.

2.5 Others

Other approaches to generating reducts from information systems have been developed and can be found in [9], [58], [67]. Among the first rough set-based approaches is the PRESET algorithm [33] which is another feature selector that uses rough set theory to rank heuristically the features, assuming a noise free binary domain. Since PRESET does not try to explore all combinations of the features, it is certain that it will fail on problems whose attributes are highly correlated. There have also been investigations into the use of different reduct quality measures [40].

select($\mathbb{C}$,$\mathbb{D}$,O,ϵ).
$\mathbb{C}$, the set of all conditional features;
$\mathbb{D}$, the set of decision features;
O, the set of objects (instances);
ϵ, reduct threshold.

(1) $R \leftarrow$ calculateCore()
(2) **while** $(\gamma_R(\mathbb{D}) < \epsilon)$
(3) $O \leftarrow O - POS_R(\mathbb{D})$ //optimization
(4) $\forall a \in \mathbb{C} - R$
(5) $v_a = |POS_{R\cup\{a\}}(D)|$
(6) $m_a = |\text{largestEquivClass}(POS_{R\cup\{a\}(D)})|$
(7) Choose a with largest $v_a * m_a$
(8) $R \leftarrow R \cup \{a\}$
(9) **return** R

Fig. 4. Zhong et al's algorithm.

In [71], a heuristic filter-based approach is presented based on rough set theory. The algorithm proposed, as reformalized in Fig. 4, begins with the core of the data set (those attributes that cannot be removed without introducing inconsistencies) and incrementally adds attributes based on a heuristic measure. Additionally, a threshold value is required as a stopping criterion to determine when a reduct candidate is "near enough" to being a reduct. On each iteration, those objects that are consistent with the current reduct candidate are removed (an optimization that can be used with RSAR). As the process starts with the core of the data set, this has to be calculated beforehand. Using the discernibility matrix for this purpose can be quite impractical for data sets of large dimensionality. However, there are other methods that can calculate the core in an efficient manner [38]. For example, this can be done by calculating the degree of dependency of the full feature set and the corresponding dependencies of the feature set minus each attribute. Those features that result in a dependency decrease are core attributes. There are also alternative methods available that allow the calculation of necessary information about the discernibility matrix without the need to perform operations directly on it [34].

Also worth mentioning are the approaches reported in [9], [67] which use genetic algorithms to discover optimal or close-to-optimal reducts. Reduct candidates are encoded as bit strings, with the value in position i set if the ith attribute is present. The fitness function depends on two parameters. The first is the number of bits set. The function penalizes those strings which have larger numbers of bits set, driving the process to find smaller reducts. The second is the number of classifiable objects given this candidate. The reduct should discern between as many objects as possible (ideally all of them).

Although this approach is not guaranteed to find minimal subsets, it may find many subsets for any given data set. It is also useful for situations where new objects are added to or old objects are removed from a data set—the reducts generated previously can be used as the initial population for the new reduct-determining process. The main drawback is the time taken to compute each bit string's fitness, which is $O(a.o^2)$, where a is the number of attributes and o the number of objects in the data set. The extent to which this hampers performance depends on several factors, including the population size.

2.6 Experimental Evaluation

In order to evaluate several of the approaches to rough set-based feature selection, an investigation into how these methods perform in terms of resulting subset optimality has been carried out. Several real and artificial data sets are used for this purpose. In particular, it is interesting to compare those methods that employ an incremental-based search strategy with those that adopt a more complex stochastic/probabilistic mechanism.

2.6.1 Dependency Degree-Based Approaches

Five techniques for finding crisp rough set reducts are tested here on 13 data sets. These techniques are: RSAR (using QUICKREDUCT), EBR (using the same search mechanism as QUICKREDUCT), GenRSAR (genetic algorithm-based), AntRSAR (ant-based), and SimRSAR (simulated annealing-based). Before the experiments are described, a few points must be made about the later three approaches, GenRSAR, AntRSAR, and SimRSAR.

GenRSAR employs a genetic search strategy in order to determine rough set reducts. The initial population consists of 100 randomly generated feature subsets, the probability of mutation and crossover set to 0.4 and 0.6 respectively, and the number of generations is set to 100. The fitness function considers both the size of subset and its evaluated suitability, and is defined as follows:

$$fitness(R) = \gamma_R(\mathbb{D}) * \frac{|\mathbb{C}| - |R|}{|\mathbb{C}|}. \tag{14}$$

AntRSAR follows the mechanism described in [24]. Here, the precomputed heuristic desirability of edge traversal is the entropy measure, with the subset evaluation performed using the rough set dependency heuristic (to guarantee that true rough set reducts are found). The number of ants used is set to the number of features, with each ant starting on a different feature. Ants construct possible solutions until they reach a rough set reduct. To avoid fruitless searches, the size of the current best reduct is used to reject those subsets whose cardinality exceed this value. Pheromone levels are set at 0.5 with a small random variation added. Levels are increased by only those ants who have found true reducts. The global search is terminated after 250 iterations, α is set to 1 and β is set to 0.1.

SimRSAR employs a simulated annealing-based feature selection mechanism [24]. The states are feature subsets, with random state mutations set to changing three features (either adding or removing them). The cost function attempts to maximize the rough set dependency (γ) while minimizing the subset cardinality. For these experiments, the cost of subset R is defined as:

$$cost(R) = \left[\frac{\gamma_{\mathbb{C}}(\mathbb{D}) - \gamma_R(\mathbb{D})}{\gamma_{\mathbb{C}}(\mathbb{D})}\right]^a + \left[\frac{|R|}{|\mathbb{C}|}\right]^b, \tag{15}$$

where a and b are defined in order to weight the contributions of dependency and subset size to the overall cost measure. In the experiments here, $a = 1$ and $b = 3$. The initial temperature of the system is estimated as $2 * |\mathbb{C}|$ and the cooling schedule is $T(t+1) = 0.93 * T(t)$.

TABLE 4
Subset Sizes Found for Five Techniques

Dataset	Features	RSAR	EBR	AntRSAR	SimRSAR	GenRSAR
M-of-N	13	8	6	6	6	6(6) 7(12)
Exactly	13	9	8	6	6	6(10) 7(10)
Exactly2	13	13	11	10	10	10(9) 11(11)
Heart	13	7	7	6(18) 7(2)	6(29) 7(1)	6(18) 7(2)
Vote	16	9	9	8	8(15) 9(15)	8(2) 9(18)
Credit	20	9	10	8(12) 9(4) 10(4)	8(18) 9(1) 11(1)	10(6) 11(14)
Mushroom	22	5	4	4	4	5(1) 6(5) 7(14)
LED	24	12	5	5(12) 6(4) 7(3)	5	6(1) 7(3) 8(16)
Letters	25	9	9	8	8	8(8) 9(12)
Derm	34	7	6	6(17) 7(3)	6(12) 7(8)	10(6) 11(14)
Derm2	34	10	10	8(3) 9(17)	8(3) 9(7)	10(2) 11(8)
WQ	38	14	14	12(2) 13(7) 14(11)	13(16) 14(4)	16
Lung	56	4	4	4	4(7) 5(12) 6(1)	6(8) 7(12)

The experiments were carried out on three data sets from [43], namely *m-of-n, exactly* and *exactly2*. The remaining data sets are from the machine learning repository [8]. Those data sets containing real-valued attributes have been discretized to allow all methods to be compared fairly.

2.6.2 Experimental Results

Table 4 shows the results of the five methods on the 13 data sets. It shows the size of reduct found for each method. RSAR and EBR produced the same subset every time, unlike AntRSAR and SimRSAR that often found different subsets and sometimes different subset cardinalities. On the whole, it appears to be the case that AntRSAR and SimRSAR outperform the other three methods. This is at the expense of the time taken to discover these reducts as can be seen in Fig. 5 (results for RSAR and EBR do not appear as they are consistently faster than the other methods). In all experiments, the rough ordering of techniques with respect to time is: RSAR $<$ EBR $\leq$ SimRSAR $\leq$ AntRSAR $\leq$ GenRSAR. AntRSAR and SimRSAR perform similarly throughout—for some data sets, AntRSAR is better (e.g., Vote) and, for others, SimRSAR is best (e.g., LED). The performance of these two methods may well be improved by fine-tuning the parameters to each individual data set.

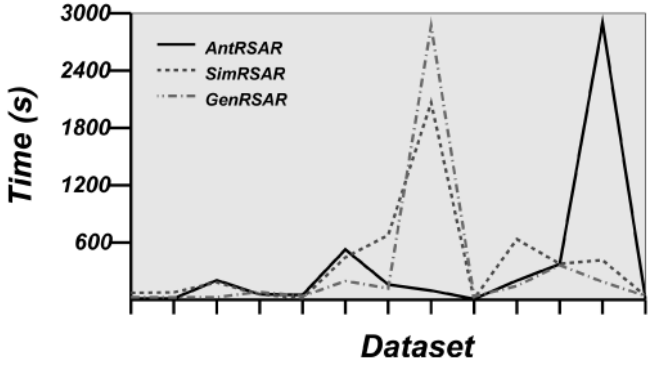

Fig. 5. Average runtimes for AntRSAR, SimRSAR, and GenRSAR.

From these results, it can be seen that even for small and medium-sized data sets, incremental hill-climbing techniques often fail to find minimal subsets. For example, RSAR is misled early in the search for the LED data set, resulting in it choosing seven extraneous features. Although this fault is due to the nonoptimality of the guiding heuristic, a perfect heuristic does not exist rendering these approaches unsuited to problems where a minimal subset is essential. However, for most real-world applications, the extent of reduction achieved via such methods is acceptable. For systems where the minimal subset is required (perhaps due to the cost of feature measurement), stochastic feature selection must be used.

2.7 Discernibility Matrix-Based Approaches

Three techniques that use the discernibility matrix to locate reducts are evaluated here on the same data sets used previously. HC is a simple hill climber that selects the next attribute based on its frequency in the clauses appearing in the discernibility matrix, following a similar strategy to that of the reduction method based on Johnson's algorithm in RSES [45]. NS follows a similar strategy to HC, but also uses information about the size of the clauses in the guiding heuristic.

Clause-based Search (CS), introduced here, performs search in a breadth-first manner. The process starts with an empty list, *Subsets*, which keeps a record of all current feature subsets. Clauses from the discernibility matrix are considered one at a time in the order of their size, with those of the smallest cardinality chosen first. When a clause is selected, the features appearing within the clause are added to every set in *Subsets*. For example, if *Subsets* contains $\{a, b\}$ and $\{c, d\}$, and the next considered clause is $\{d \vee e\}$, then each appearing attribute is added. The *Subsets* list will now contain $\{a, b, d\}$, $\{a, b, e\}$, $\{c, d\}$, and $\{c, d, e\}$. This guarantees that each set in *Subsets* satisfies all the clauses that have been encountered so far. If one of these subsets satisfies all clauses, the algorithm terminates as a

TABLE 5
Subset Sizes Found for Discernibility Matrix-Based Techniques

Dataset	Features	HC	NS	CS
M-of-N	13	6	6	6
Exactly	13	6	6	6
Exactly2	13	10	10	10
Heart	13	6	6	6
Vote	16	8	8	8
Credit	20	10	10	8
Mushroom	22	4	4	4
LED	24	5	5	5
Letters	25	9	10	8
Derm	34	6	6	6
Derm2	34	9	9	8
WQ	38	14	13	12
Lung	56	4	4	4

reduct has been found. If not, then the process continues by selecting the next clause and adding these features. This process will result in a minimal subset, but has an exponential time and space complexity.

The results of the application of these three methods to the 13 data sets can be found in Table 5. HC and NS perform similarly throughout, differing only in their results for the Letters and WQ data sets. CS will always find the smallest valid feature subset, though is too costly to apply to larger data sets in its present form. On the whole, all three methods perform as well as or better than the dependency-based methods. HC, NS, and CS all require the calculation of the discernibility matrix beforehand, however, there are methods to avoid such computation [34].

The utility of rough set-selected subsets in classification has been shown in [50], where several dimensionality reducers were used for neural network-based image classification. The reduct produced by RSAR resulted in the lowest classification error of the trained network, even surpassing PCA. The features selected by the rough set method also correspond to those chosen by experts in determining manual classifications.

3 Fuzzy Rough Attribute Reduction

All rough set-based FS methods previously described can only operate effectively with data sets containing discrete values. As most data sets contain real-valued attributes, it is necessary to perform a discretization step beforehand. Boolean discretization can be very difficult to match human understanding of the respective domains, however. To reduce this difficulty, discretization can be implemented by a standard fuzzification technique [51]. Nevertheless, membership degrees of attribute values to fuzzy sets are typically not exploited in the process of dimensionality reduction. This is counterintuitive. By using *fuzzy-rough* sets [17], [36], [64], [25], it is possible to use this information to better guide feature selection. The approach presented here differs significantly from those such as [59] that are concerned with discrete but inconsistent data. The novel fuzzy-rough method and grouping mechanism presented here are concerned with real valued attributes with corresponding fuzzifications.

3.1 Fuzzy Equivalence Classes

In the same way that crisp equivalence classes are central to rough sets, *fuzzy* equivalence classes are central to the fuzzy-rough set approach [17]. In classification applications, for example, this means that the decision values and the conditional values may all be fuzzy. The concept of crisp equivalence classes can be extended by the inclusion of a fuzzy similarity relation S on the universe, which determines the extent to which two elements are similar in S [21]. The usual properties of reflexivity ($\mu_S(x,x) = 1$), symmetry ($\mu_S(x,y) = \mu_S(y,x)$), and transitivity ($\mu_S(x,z) \geq \mu_S(x,y) \wedge \mu_S(y,z)$, where $\wedge$ is a t-norm) hold.

Using the fuzzy similarity relation S, the fuzzy equivalence class $[x]_S$ for objects close to x can be defined:

$$\mu_{[x]_S}(y) = \mu_S(x,y). \tag{16}$$

The following axioms should hold for a fuzzy equivalence class $F = [x]_S$ [21]:

- $\exists x$, $\mu_F(x) = 1$,
- $\mu_F(x) \wedge \mu_S(x,y) \leq \mu_F(y)$, and
- $\mu_F(x) \wedge \mu_F(y) \leq \mu_S(x,y)$.

The first axiom corresponds to the requirement that an equivalence class is nonempty. The second axiom states that elements in y's neighborhood are in the equivalence class of y. The final axiom states that any two elements in F are related via S. Obviously, this definition degenerates to the normal definition of equivalence classes when S is nonfuzzy.

The family of normal fuzzy sets produced by a fuzzy partitioning of the universe of discourse can play the role of fuzzy equivalence classes [17]. Consider the crisp partitioning $\mathbb{U}/Q = \{\{1,3,6\},\{2,4,5\}\}$. This contains two equivalence classes ($\{1,3,6\}$ and $\{2,4,5\}$) that can be thought of as degenerated fuzzy sets, with those elements belonging to the class possessing a membership of one, zero otherwise. For the first class, for instance, the objects 2, 4, and 5 have a membership of zero. Extending this to the case of fuzzy equivalence classes is straightforward: Objects can be allowed to assume membership values, with respect to any given class, in the interval [0,1]. $\mathbb{U}/Q$ is not restricted to crisp partitions only; fuzzy partitions are equally acceptable.

3.2 Fuzzy Lower and Upper Approximations

From the literature, the fuzzy P-lower and P-upper approximations are defined as [17]:

$$\mu_{\underline{P}X}(F_i) = inf_x max\{1 - \mu_{F_i}(x), \mu_X(x)\} \quad \forall i, \tag{17}$$

$$\mu_{\overline{P}X}(F_i) = sup_x min\{\mu_{F_i}(x), \mu_X(x)\} \quad \forall i, \tag{18}$$

where F_i denotes a fuzzy equivalence class belonging to $\mathbb{U}/P$. Note that, although the universe of discourse in attribute reduction is finite, this is not the case in general, hence, the use of sup and inf. These definitions diverge a little from the crisp upper and lower approximations, as the memberships of individual objects to the approximations

are not explicitly available. As a result of this, the fuzzy lower and upper approximations are herein redefined as:

$$\mu_{\underline{P}X}(x) = \sup_{F \in \mathbb{U}/P} min(\mu_F(x), \inf_{y \in \mathbb{U}} max\{1 - \mu_F(y), \mu_X(y)\}), \tag{19}$$

$$\mu_{\overline{P}X}(x) = \sup_{F \in \mathbb{U}/P} min(\mu_F(x), \sup_{y \in \mathbb{U}} min\{\mu_F(y), \mu_X(y)\}). \tag{20}$$

In implementation, not all $y \in \mathbb{U}$ are needed to be considered—only those where $\mu_F(y)$ is nonzero, i.e., where object y is a fuzzy member of (fuzzy) equivalence class F.

The tuple $< \underline{P}X, \overline{P}X >$ is called a fuzzy-rough set. It can be seen that these definitions degenerate to traditional rough sets when all equivalence classes are crisp. It is useful to think of the crisp lower approximation as characterized by the following membership function:

$$\mu_{\underline{P}X}(x) = \begin{cases} 1, & x \in F, \ F \subseteq X \\ 0, & otherwise \end{cases} \tag{21}$$

This states that an object x belongs to the P-lower approximation of X if it belongs to an equivalence class that is a subset of X. The behavior of the fuzzy lower approximation must be exactly that of the crisp definition for crisp situations. This is indeed the case as the fuzzy lower approximation may be rewritten as

$$\mu_{\underline{P}X}(x) = \sup_{F \in \mathbb{U}/P} min(\mu_F(x), \inf_{y \in \mathbb{U}} \{\mu_F(y) \rightarrow \mu_X(y)\}), \tag{22}$$

where $\rightarrow$ denotes the fuzzy implication operator. In the crisp case, $\mu_F(x)$ and $\mu_X(x)$ will take values from $\{0, 1\}$. Hence, it is clear that the only time $\mu_{\underline{P}X}(x)$ will be zero is when at least one object in its equivalence class F fully belongs to F but not to X. This is exactly the same as the definition for the crisp lower approximation. Similarly, the definition for the P-upper approximation can be established to make sense in being the generalization of the crisp definition.

3.3 Fuzzy-Rough Reduction Process

Fuzzy RSAR (abbreviated FRAR hereafter) builds on the notion of the fuzzy lower approximation to enable reduction of data sets containing real-valued attributes. As will be shown, the process becomes identical to the traditional approach when dealing with nominal well-defined attributes. This feature selection method has been used in Web categorization [25] and complex systems monitoring [52].

The crisp positive region in traditional rough set theory is defined as the union of the lower approximations. By the extension principle, the membership of an object $x \in \mathbb{U}$, belonging to the fuzzy positive region can be defined by

$$\mu_{POS_P(Q)}(x) = \sup_{X \in \mathbb{U}/Q} \mu_{\underline{P}X}(x). \tag{23}$$

Object x will not belong to the positive region only if the equivalence class it belongs to is not a constituent of the positive region. This is equivalent to the crisp version where objects belong to the positive region only if their underlying equivalence class does so.

Using the definition of the fuzzy positive region, the new dependency function can be defined as follows:

$$\gamma'_P(Q) = \frac{|\mu_{POS_P(Q)}(x)|}{|\mathbb{U}|} = \frac{\sum_{x \in \mathbb{U}} \mu_{POS_P(Q)}(x)}{|\mathbb{U}|}. \tag{24}$$

As with crisp rough sets, the dependency of Q on P is the proportion of objects that are discernible out of the entire data set. In the present approach, this corresponds to determining the fuzzy cardinality of $\mu_{POS_P(Q)}(x)$ divided by the total number of objects in the universe.

The definition of dependency degree covers the crisp case as its specific instance. This can be easily shown by recalling the definition of the crisp dependency degree given in (24). If a function $\mu_{POS_P(Q)}(x)$ is defined which returns 1 if the object x belongs to the positive region, 0 otherwise, then the above definition may be rewritten as:

$$\gamma_P(Q) = \frac{\sum_{x \in \mathbb{U}} \mu_{POS_P(Q)}(x)}{|\mathbb{U}|}, \tag{25}$$

which is identical to (24).

If the fuzzy-rough reduction process is to be useful, it must be able to deal with multiple attributes, finding the dependency between various subsets of the original attribute set. For example, it may be necessary to be able to determine the degree of dependency of the decision attribute(s) with respect to $P = \{a, b\}$. In the crisp case, $\mathbb{U}/P$ contains sets of objects grouped together that are indiscernible according to both attributes a and b. In the fuzzy case, objects may belong to many equivalence classes, so the Cartesian product of $\mathbb{U}/IND(\{a\})$ and $\mathbb{U}/IND(\{b\})$ must be considered in determining $\mathbb{U}/P$. In general,

$$\mathbb{U}/P = \otimes\{a \in P : \mathbb{U}/IND(\{a\})\} \tag{26}$$

Each set in $\mathbb{U}/P$ denotes an equivalence class. For example, if $P = \{a, b\}$, $\mathbb{U}/IND(\{a\}) = \{N_a, Z_a\}$, and $\mathbb{U}/IND(\{b\}) = \{N_b, Z_b\}$, then

$$\mathbb{U}/P = \{N_a \cap N_b, N_a \cap Z_b, Z_a \cap N_b, Z_a \cap Z_b\}.$$

The extent to which an object belongs to such an equivalence class is therefore calculated by using the conjunction of constituent fuzzy equivalence classes, say F_i, $i = 1, 2, \ldots, n$:

$$\mu_{F_1 \cap \ldots \cap F_n}(x) = min(\mu_{F_1}(x), \mu_{F_2}(x), \ldots, \mu_{F_n}(x)). \tag{27}$$

3.4 Reduct Computation

In conventional RSAR, a reduct is defined as a subset R of the attributes which have the same information content as the full attribute set A. In terms of the dependency function, this means that the values $\gamma(R)$ and $\gamma(A)$ are identical and equal to 1 if the data set is consistent. However, in the fuzzy-rough approach, this is not necessarily the case as the uncertainty encountered when objects belong to many fuzzy equivalence classes results in a reduced total dependency.

A possible way of combatting this would be to determine the degree of dependency of the full attribute set and use this as the denominator (for normalization rather than $|\mathbb{U}|$), allowing γ' to reach 1. With these issues in mind, a new QUICKREDUCT algorithm has been developed as given in

FRQUICKREDUCT($\mathbb{C}$,$\mathbb{D}$).
$\mathbb{C}$, the set of all conditional features;
$\mathbb{D}$, the set of decision features.

(1) $R \leftarrow \{\}$; $\gamma'_{best} = 0$; $\gamma'_{prev} = 0$
(2) **do**
(3) $T \leftarrow R$
(4) $\gamma'_{prev} = \gamma'_{best}$
(5) $\forall x \in (\mathbb{C} - R)$
(6) **if** $\gamma'_{R \cup \{x\}}(\mathbb{D}) > \gamma'_{T}(\mathbb{D})$
(7) $T \leftarrow R \cup \{x\}$
(8) $\gamma'_{best} = \gamma'_{T}(\mathbb{D})$
(9) $R \leftarrow T$
(10) **until** $\gamma'_{best} == \gamma'_{prev}$
(11) **return** R

Fig. 6. The fuzzy-rough QUICKREDUCT algorithm.

Fig. 6. It employs the new dependency function γ' to choose which attributes to add to the current reduct candidate in the same way as the original QUICKREDUCT process. The algorithm terminates when the addition of any remaining attribute does not increase the dependency (such a criterion could be used with the original QUICKREDUCT algorithm). As with the original QUICKREDUCT algorithm, for a dimensionality of n, the worst-case data set will result in $(n^2 + n)/2$ evaluations of the dependency function. However, as both fuzzy and crisp RSAR is used for dimensionality reduction prior to any involvement of an application system which will employ those attributes belonging to the resultant reduct, this potentially costly operation has no negative impact upon the runtime efficiency of the system.

Note that it is also possible to reverse the search process; that is, start with the full set of attributes and incrementally remove the least informative attributes. This process continues until no more attributes can be removed without reducing the total number of discernible objects in the data set.

3.5 Fuzzy RSAR Example

To illustrate the operation of fuzzy RSAR, an example data set is given in Fig. 7. In crisp RSAR, the data set would be discretized using the nonfuzzy sets. However, in the new approach, membership degrees are used in calculating the fuzzy lower approximations and fuzzy positive regions. To begin with, the fuzzy-rough QUICKREDUCT algorithm initializes the potential reduct (i.e., the current best set of attributes) to the empty set.

Using the fuzzy sets defined in Fig. 7 (for all conditional attributes for illustrative simplicity), and setting $A = \{a\}$, $B = \{b\}$, $C = \{c\}$, and $Q = \{q\}$, the following equivalence classes are obtained:

$$\mathbb{U}/A = \{N_a, Z_a\}$$
$$\mathbb{U}/B = \{N_b, Z_b\}$$
$$\mathbb{U}/C = \{N_c, Z_c\}$$
$$\mathbb{U}/Q = \{\{1,3,6\},\{2,4,5\}\}.$$

The first step is to calculate the lower approximations of the sets A, B, and C. For straightforwardness, only the calculations involving A are demonstrated here; that is, using A to approximate Q. For the first decision equivalence class $X = \{1,3,6\}$, $\mu_{\underline{A}\{1,3,6\}}(x)$ is calculated:

$$\mu_{\underline{A}\{1,3,6\}}(x) = \sup_{F \in \mathbb{U}/A} min(\mu_F(x), \inf_{y \in \mathbb{U}} max\{1 - \mu_F(y), \mu_{\{1,3,6\}}(y)\}).$$

Considering the first fuzzy equivalence class of A, N_a:

$$min(\mu_{N_a}(x), \inf_{y \in \mathbb{U}} max\{1 - \mu_{N_a}(y), \mu_{\{1,3,6\}}(y)\}).$$

For object 2, this can be calculated as follows:

$$min(0.8, inf\{1, 0.2, 1, 1, 1, 1\}) = 0.2$$

Similarly, for Z_a,

$$min(0.2, inf\{1, 0.8, 1, 0.6, 0.4, 1\} = 0.2.$$

Thus,

$$\mu_{\underline{A}\{1,3,6\}}(2) = 0.2.$$

Calculating the A-lower approximation of $X = \{1,3,6\}$ for every object gives

$$\mu_{\underline{A}\{1,3,6\}}(1) = 0.2 \quad \mu_{\underline{A}\{1,3,6\}}(2) = 0.2$$
$$\mu_{\underline{A}\{1,3,6\}}(3) = 0.4 \quad \mu_{\underline{A}\{1,3,6\}}(4) = 0.4$$
$$\mu_{\underline{A}\{1,3,6\}}(5) = 0.4 \quad \mu_{\underline{A}\{1,3,6\}}(6) = 0.4.$$

The corresponding values for $X = \{2,4,5\}$ can also be determined this way. Using these values, the fuzzy positive region for each object can be calculated via using

$$\mu_{POS_A(Q)}(x) = \sup_{X \in \mathbb{U}/Q} \mu_{\underline{A}X}(x).$$

This results in:

$$\mu_{POS_A(Q)}(1) = 0.2 \quad \mu_{POS_A(Q)}(2) = 0.2$$
$$\mu_{POS_A(Q)}(3) = 0.4 \quad \mu_{POS_A(Q)}(4) = 0.4$$
$$\mu_{POS_A(Q)}(5) = 0.4 \quad \mu_{POS_A(Q)}(6) = 0.4.$$

Object	a	b	c	q
1	-0.4	-0.3	-0.5	no
2	-0.4	0.2	-0.1	yes
3	-0.3	-0.4	-0.3	no
4	0.3	-0.3	0	yes
5	0.2	-0.3	0	yes
6	0.2	0	0	no

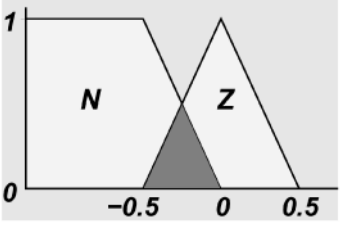

Fig. 7. Data set and corresponding fuzzy sets.

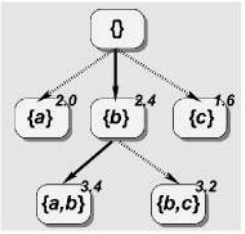

Fig. 8. Path taken by the fuzzy-rough QUICKREDUCT algorithm.

It is a coincidence here that $\mu_{POS_A(Q)}(x) = \mu_{\underline{A}\{1,3,6\}}(x)$ for this example. The next step is to determine the degree of dependency of Q on A:

$$\gamma'_A(Q) = \frac{\sum_{x \in U} \mu_{POS_A(Q)}(x)}{|U|} = 2/6.$$

Similarly, calculating for B and C gives:

$$\gamma'_B(Q) = \frac{2.4}{6}, \quad \gamma'_C(Q) = \frac{1.6}{6}.$$

From this, it can be seen that attribute b will cause the greatest increase in dependency degree. This attribute is chosen and added to the potential reduct. The process iterates and the two dependency degrees calculated are

$$\gamma'_{\{a,b\}}(Q) = \frac{3.4}{6}, \quad \gamma'_{\{b,c\}}(Q) = \frac{3.2}{6}.$$

Adding attribute a to the reduct candidate causes the larger increase of dependency, so the new candidate becomes $\{a, b\}$. Last, attribute c is added to the potential reduct:

$$\gamma'_{\{a,b,c\}}(Q) = \frac{3.4}{6}.$$

As this causes no increase in dependency, the algorithm stops and outputs the reduct $\{a, b\}$ (see Fig. 8). The data set can now be reduced to only those attributes appearing in the reduct. When crisp RSAR is performed on this data set (after using the same fuzzy sets to discretize the real-valued attributes), the reduct generated is $\{a, b, c\}$, i.e., the full conditional attribute set. Unlike crisp RSAR, the true minimal reduct was found using the information on degrees of membership. It is clear from this example alone that the information lost by using crisp RSAR can be important when trying to discover the smallest reduct from a data set.

3.6 Rough Set-Based Feature Grouping

By its definition, the degree of dependency measure (whether using crisp or fuzzy-rough sets) always lies in the range [0,1], with 0 indicating no dependency and 1 indicating total dependency. For example, two subsets of the conditional attributes in a data set may have the following dependency degrees:

$$\gamma'_{\{a,b,c\}}(\mathbb{D}) = 0.54, \quad \gamma'_{\{a,c,d\}}(\mathbb{D}) = 0.52.$$

In traditional rough sets, it would be said that the attribute set $\{a, b, c\}$ has a higher dependency value than $\{a, c, d\}$ and so would make the better candidate to produce a minimal reduct. This may not be the case when considering real data sets that contain noise and other discrepancies. In fact, it is possible that $\{a, c, d\}$ is the best candidate for this and other unseen related data sets. By fuzzifying the output values of the dependency function, this problem may be successfully tackled. In addition to this, attributes may be *grouped* at stages in the selection process depending on their dependency label, speeding up the reduct search.

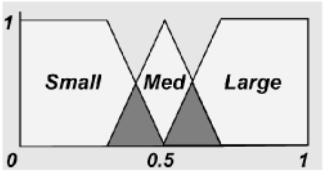

Fig. 9. Possible fuzzification of dependency.

In order to achieve this, several fuzzy sets must be defined over the dependency range (for example, Fig. 9). This leads to the next problem area: How are these sets to be defined? There is also the problem of how many fuzzy sets should be used to produce the most useful results. Initially, these may be defined beforehand by an expert and refined through experimentation. However, to fit in with the rough set ideology, it would be interesting to investigate how to automatically generate these sets purely from the data set itself (perhaps using a clustering method). For the time being, it is assumed that these fuzzy sets have already been defined.

The goal of RSAR and FRAR is to find a (possibly minimal) subset of the conditional attributes for which the degree of dependency is at a maximum (ideally, the value 1). In the case of fuzzy equivalence classes, where an element of uncertainty is introduced, the maximum degree of dependency may be substantially less than this. In fact, the maximum dependency for different data sets may be quite different due to differing levels of uncertainty. The maximum for data set A may be 0.9 whereas, for data set B, the maximum may be only 0.2. Given a degree of dependency of 0.19, for data set A, this is quite a small value, but for data set B, this is quite large, so some way of scaling the dependency value depending on the data set is required. The following is one potential way of achieving this for a subset P of all conditional attributes $\mathbb{C}$:

$$\gamma''_P(\mathbb{D}) = \frac{\gamma'_P(\mathbb{D})}{\gamma'_{\mathbb{C}}(\mathbb{D})}.$$

In the example above, the scaled dependency degree for data set A is now 0.21 (which fuzzifies to $Small$) and for data set B is 0.95 (which fuzzifies to $Large$). However, a further problem is encountered as the search for a reduct nears its conclusion. In this situation, almost all of the dependency values are mapped to $Large$ due to their underlying closeness in value. This means that too large a group of attributes will be selected every time. Additionally, if the data is noisy, it may be the case that $\gamma''_P(\mathbb{D}) > 1$ as the dependency degree of the full set of conditional attributes may be greater than that of a particular attribute subset. An alternative scaling approach to combat both of these problems is to use the extreme values at each level of

FUZZYQUICKREDUCT($\mathbb{C}$,$\mathbb{D}$).
$\mathbb{C}$, the set of all conditional attributes;
$\mathbb{D}$, the set of decision attributes.

(1) $R \leftarrow \{\}$; $\gamma'_{best} = 0$; $\gamma'_{prev} = 0$
(2) **do**
(3) $Cands \leftarrow \{\}$
(4) $\gamma'_{prev} = \gamma'_{best}$
(5) $\gamma'_{high} = 0$; $\gamma'_{low} = 1$
(5) $\forall x \in (\mathbb{C} - R)$
(6) $T \leftarrow R \cup \{x\}$
(7) $Cands \leftarrow Cands \cup (x, \gamma'_T(\mathbb{D}))$
(8) **if** $\gamma'_T(\mathbb{D}) > \gamma'_{high}$
(9) $\gamma'_{high} = \gamma'_T(\mathbb{D})$
(10) **else if** $\gamma'_T(\mathbb{D}) < \gamma'_{low}$
(11) $\gamma'_{low} = \gamma'_T(\mathbb{D})$
(12) $Cands \leftarrow \text{scale}(Cands, \gamma'_{high}, \gamma'_{low})$
(13) $R \leftarrow R \cup \text{selectFeatures}(Cands)$
(14) $\gamma'_{best} = \gamma'_R(\mathbb{D})$
(15) **until** $\gamma'_{best} == \gamma'_{prev}$
(16) **return** R

Fig. 10. The new fuzzy-rough QUICKREDUCT algorithm with fuzzy dependencies.

search. As soon as the reduct candidates have been evaluated, the highest and lowest dependencies ($\gamma'_{high}(\mathbb{D})$ and $\gamma'_{low}(\mathbb{D})$) are used as follows to scale the dependency degree of subset P:

$$\gamma''_P(\mathbb{D}) = \frac{\gamma'_P(\mathbb{D}) - \gamma'_{low}(\mathbb{D})}{\gamma'_{high}(\mathbb{D}) - \gamma'_{low}(\mathbb{D})}.$$

By this method, the attribute subset with the highest dependency value will have a scaled dependency ($\gamma''_P(\mathbb{D})$) of 1. The subset with the lowest will have a scaled dependency of 0. In so doing, the definition of the fuzzy sets need not be changed for different data sets; one definition should be applicable to all.

The next question to address is how to handle those scaled dependencies that fall at the boundaries. For example, a value may partially belong to both *Small* and *Medium*. A simple strategy is to choose the single fuzzy label with the highest membership value. However, this loses the potentially useful information of dual fuzzy set membership. Another strategy is to take both labels as valid, considering both possibilities within the feature selection process. If, for example, a dependency value lies within the labels *Small* and *Medium* then it is considered to belong to both groups.

The new fuzzy-rough QUICKREDUCT algorithm (FQR) which employs scaling and fuzzy dependencies can be seen in Fig. 10. In this algorithm, $Cands$ contains sets of attributes and their corresponding dependency degrees when added to the current reduct candidate. Once each remaining attribute has been evaluated, the dependencies are scaled according to γ'_{high} and γ'_{low}. Next, the decision is made on which feature(s) to add to the current reduct. In the previous fuzzy-rough QUICKREDUCT algorithm, this would amount to selecting the feature providing the highest gain in dependency degree. Here, other strategies may be employed; for example, attributes may be selected individually or in groups. This is discussed in more detail next.

Note that, in addition to applying this method to fuzzy-rough attribute reduction, it may also be applied to crisp RSAR. Given a data set containing crisp values, the dependency values may be fuzzified similarly (with scaling) so that groups of attributes may be selected at one time. The algorithm for this is exactly the same as the one given in (10), except the dependency function used is now based on crisp rough sets.

3.7 Selection Strategies

When using fuzzy degrees of dependency, it is possible to change strategy at any stage of the attribute selection process. The main distinction to make in the set of possible strategies is whether features are chosen individually or in groups.

3.7.1 Individuals

In this subset of strategies, attributes are chosen one at a time in a similar fashion to that of FRAR. However, the choice of attribute depends on its corresponding linguistic label(s) obtained from the dependency degree. In the example, fuzzification of dependency given in Fig. 9, attributes are grouped into the categories $Small$, $Medium$, and $Large$. A representative attribute of the required label can be chosen randomly or based on the extent to which the attribute belongs to the fuzzy set itself. Those individual attributes lying on set boundaries are assigned both fuzzy labels. Other issues include which particular group of attributes to consider. Intuitively, it would seem most appropriate to consider those belonging to the $Large$ group only, however, it may be worthwhile investigating $Small$ and $Medium$-grouped attributes at different stages of the search process.

3.7.2 Grouping

To speed up the reduct search process, many attributes may be added to a reduct candidate at once, according to their label. For instance, selecting only those attributes considered to be $Large$ would appear to be a suitable strategy. It may also be beneficial to add different groups of attributes at various stages of the search. To include diversity, crossgroup selection is a method that picks representative attributes from each fuzzy label and adds them to the reduct candidate. Again, strategies may be changed during search; for example, it might be worthwhile using the crossgroup strategy first, followed by selecting $Large$-grouped attributes later.

One problem encountered in grouping attributes in this way is that, in later stages, there are sometimes too many attributes in the required label. Therefore, it is usually best to revert to individual selection when this becomes a problem, making the search more accurate. Initial results of the application of this feature grouping technique to complex systems monitoring can be found in [23]. FQR performs at least as well as FRAR in this study.

4 CONCLUSION

Feature selection seeks to reduce data while retaining semantics by selecting attributes as opposed to transforming them. This aspect is particularly useful when feature selection precedes other processes that require the original feature meanings to be intact, for example, rule induction where rules may need to be human-readable. This review

focused on some of the recent developments in rough set theory for the purpose of feature selection.

Several approaches to discovering rough set reducts were experimentally evaluated and compared. The results highlighted the shortcomings of conventional hill-climbing approaches to feature selection. These techniques often fail to find minimal data reductions. Some guiding heuristics are better than others for this, but, as no perfect heuristic exists, there can be no guarantee of optimality. From the experimentation, it appears that the entropy-based measure is a more useful hill-climbing heuristic than the rough set-based one. However, the entropy measure is a more costly operation than that of dependency evaluation which may be an important factor when processing large data sets. Due to the failure of hill-climbing methods and the fact that exhaustive searches are not feasible for even medium-sized data sets, stochastic approaches provide a promising feature selection mechanism.

Conventional rough set methods are unable to deal with real-valued attributes effectively. This prompted research into the use of fuzzy-rough sets for feature selection. Additionally, the new direction in feature selection, *feature grouping*, was highlighted. It was shown how fuzzifying a particular evaluation function, the rough set dependency degree, can lead to group and individual selection based on linguistic labels—more closely resembling human reasoning. In fact, this can be applied to most FS algorithms that use an evaluation function that returns values in $[0, 1]$. Choosing grouped features instead of individuals also decreases the time taken to reach potentially optimal subsets.

REFERENCES

[1] H. Almuallim and T.G. Dietterich, "Learning with Many Irrelevant Features," *Proc. Ninth Nat'l Conf. Artificial Intelligence,* pp. 547-552, 1991.

[2] "Rough Sets and Current Trends in Computing," *Proc. Third Int'l Conf.,* J.J. Alpigini, J.F. Peters, J. Skowronek, and N. Zhong, eds., 2002.

[3] J. Bazan, A. Skowron, and P. Synak, "Dynamic Reducts as a Tool for Extracting Laws from Decision Tables," *Proc. Eighth Symp. Methodologies for Intelligent Systems,* Z.W. Ras and M. Zemankova, eds., pp. 346-355, 1994.

[4] J. Bazan, "A Comparison of Dynamic and Non-Dynamic Rough Set Methods for Extracting Laws from Decision Tables," *Rough Sets in Knowledge Discovery,* L. Polkowski and A. Skowron, eds., pp. 321-365, Physica - Verlag, 1998.

[5] T. Beaubouef, F.E. Petry, and G. Arora, "Information Measures for Rough and Fuzzy Sets and Application to Uncertainty in Relational Databases," *Rough-Fuzzy Hybridization: A New Trend in Decision Making,* 1999.

[6] M.J. Beynon, "An Investigation of β-Reduct Selection within the Variable Precision Rough Sets Model," *Proc. Second Int'l Conf. Rough Sets and Current Trends in Computing (RSCTC 2000),* pp. 114-122, 2000.

[7] M.J. Beynon, "Reducts within the Variable Precision Rough Sets Model: A Further Investigation," *European J. Operational Research,* vol. 134, no. 3, pp. 592-605, 2001.

[8] C.L. Blake and C.J. Merz UCI Repository of Machine Learning Databases, University of California at Irvine, 1998, http://www.ics.uci.edu/~mlearn/.

[9] A.T. Bjorvand and J. Komorowski, "Practical Applications of Genetic Algorithms for Efficient Reduct Computation," *Proc. 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics,* A. Sydow, ed., vol. 4, pp. 601-606, 1997.

[10] A. Chouchoulas and Q. Shen, "Rough Set-Aided Keyword Reduction for Text Categorisation," *Applied Artificial Intelligence,* vol. 15, no. 9, pp. 843-873, 2001.

[11] A. Chouchoulas, J. Halliwell, and Q. Shen, "On the Implementation of Rough Set Attribute Reduction," *Proc. 2002 UK Workshop Computational Intelligence,* pp. 18-23, 2002.

[12] M. Dash and H. Liu, "Feature Selection for Classification," *Intelligent Data Analysis,* vol. 1, no. 3, 1997.

[13] P. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach.* Prentice Hall, 1982.

[14] J. Dong, N. Zhong, and S. Ohsuga, "Using Rough Sets with Heuristics for Feature Selection," *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, Proc. Seventh Int'l Workshop (RSFDGrC '99),* pp. 178-187, 1999.

[15] G. Drwal and A. Mrókek, "System RClass—Software Implementation of the Rough Classifier," *Proc. Seventh Int'l Symp. Intelligent Information Systems,* pp. 392-395, 1998.

[16] G. Drwal, "Rough and Fuzzy-Rough Classification Methods Implemented in RClass System," *Proc. Second Int'l Conf. Rough Sets and Current Trends in Computing (RSCTC 2000),* pp. 152-159, 2000.

[17] D. Dubois and H. Prade, "Putting Rough Sets and Fuzzy Sets Together," *Intelligent Decision Support,* pp. 203-232, 1992.

[18] "Rough Set Data Analysis," I. Düntsch and G. Gediga, eds., *Encyclopedia of Computer Science and Technology,* A. Kent and J.G. Williams, eds., pp. 281-301, 2000.

[19] I. Düntsch and G. Gediga, *Rough Set Data Analysis: A Road to Non-Invasive Knowledge Discovery.* Bangor: Methodos, 2000.

[20] *A Brief Introduction to Rough Sets,* EBRSC, Copyright 1993, information available at http://cs.uregina.ca/~roughset/rs.intro.txt.

[21] U. Höhle, "Quotients with Respect to Similarity Relations," *Fuzzy Sets and Systems,* vol. 27, pp. 31-44, 1988.

[22] J. Jelonek, K. Krawiec, and R. Slowinski, "Rough Set Reduction of Attributes and Their Domains for Neural Networks," *Computational Intelligence 11,* pp. 339-347, 1995.

[23] R. Jensen and Q. Shen, "Using Fuzzy Dependency-Guided Attribute Grouping in Feature Selection," *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Proc. Ninth Int'l Conf. (RSFDGrC 2003),* pp. 250-255, 2003.

[24] R. Jensen and Q. Shen, "Finding Rough Set Reducts with Ant Colony Optimization," *Proc. 2003 UK Workshop Computational Intelligence,* pp. 15-22, 2003.

[25] R. Jensen and Q. Shen, "Fuzzy-Rough Attribute Reduction with Application to Web Categorization," *Fuzzy Sets and Systems,* vol. 141, no. 3, pp. 469-485, 2004.

[26] K. Kira and L.A. Rendell, "The Feature Selection Problem: Traditional Methods and a New Algorithm," *Proc. Ninth Nat'l Conf. Artificial Intelligence,* pp. 129-134, 1992.

[27] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron, "Rough Sets: A Tutorial," *Rough-Fuzzy Hybridization: A New Trend in Decision Making,* pp. 3-98, 1999.

[28] M. Kryszkiewicz, "Maintenance of Reducts in the Variable Precision Rough Sets Model," ICS Research Report 31/94, Warsaw Univ. of Technology, 1994.

[29] P. Langley, "Selection of Relevant Features in Machine Learning," *Proc. AAAI Fall Symp. Relevance,* pp. 1-5, 1994.

[30] *Feature Extraction, Construction and Selection: A Data Mining Perspective (Kluwer International Series in Engineering & Computer Science),* H. Liu and H. Motoda, eds. Kluwer Academic Publishers, 1998.

[31] A.J. Miller, *Subset Selection in Regression.* Chapman and Hall, 1990.

[32] T. Mitchell, *Machine Learning.* McGraw-Hill, 1997.

[33] M. Modrzejewski, "Feature Selection Using Rough Sets Theory," *Proc. 11th Int'l Conf. Machine Learning,* pp. 213-226, 1993.

[34] S.H. Nguyen and H.S. Nguyen, "Some Efficient Algorithms for Rough Set Methods," *Proc. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems,* pp. 1451-1456, 1996.

[35] H.S. Nguyen and A. Skowron, "Boolean Reasoning for Feature Extraction Problems," *Proc. Int'l Symp. Methodologies for Intelligent Systems (ISMIS),* pp. 117-126, 1997.

[36] *Rough-Fuzzy Hybridization: A New Trend in Decision Making,* S.K. Pal and A. Skowron, eds. Springer Verlag, 1999.

[37] Z. Pawlak, "Rough Sets," *Int'l J. Computer and Information Sciences,* vol. 11, no. 5, pp. 341-356, 1982.

[38] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data.* Kluwer Academic Publishing, 1991.
[39] Z. Pawlak and A. Skowron, "Rough Membership Functions," *Advances in the Dempster-Shafer Theory of Evidence,* R. Yager, M. Fedrizzi, and J. Kacprzyk, eds., pp. 251-271, 1994.
[40] "Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems," *Studies in Fuzziness and Soft Computing,* L. Polkowski, T.Y. Lin, and S. Tsumoto, eds., vol. 56, Physica-Verlag, 2000.
[41] L. Polkowski, "Rough Sets: Mathematical Foundations," *Advances in Soft Computing,* Physica Verlag, 2002.
[42] J.R. Quinlan, "C4.5: Programs for Machine Learning," *The Morgan Kaufmann Series in Machine Learning,* Morgan Kaufmann Publishers, 1993.
[43] B. Raman and T.R. Ioerger, "Instance-Based Filter for Feature Selection," *J. Machine Learning Research 1,* pp. 1-23, 2002.
[44] The ROSETTA homepage, http://rosetta.lcb.uu.se/general/, 2004.
[45] RSES: Rough Set Exploration System, http://logic.mimuw.edu.pl/~rses, 2004.
[46] J.C. Schlimmer, "Efficiently Inducing Determinations—A Complete and Systematic Search Algorithm that Uses Optimal Pruning," *Proc. Int'l Conf. Machine Learning,* pp. 284-290, 1993.
[47] R. Setiono and H. Liu, "Neural Network Feature Selector," *IEEE Trans. Neural Networks,* vol. 8, no. 3, pp. 645-662, 1997.
[48] H. Sever, V.V. Raghavan, and T.D. Johnsten, "The Status of Research on Rough Sets for Knowledge Discovery in Databases," *Proc. ICNPAA-98: Second Int'l Conf. Nonlinear Problems in Aviation and Aerospace,* 1998.
[49] G. Shafer, *A Mathematical Theory of Evidence.* Princeton Univ. Press, 1976.
[50] C. Shang and Q. Shen, "Rough Feature Selection for Neural Network Based Image Classification," *Int'l J. Image Graphics,* vol. 2, no. 4, pp. 541-556, 2002.
[51] Q. Shen and A. Chouchoulas, "A Fuzzy-Rough Approach for Generating Classification Rules," *Pattern Recognition,* vol. 35, no. 11, pp. 341-354, 2002.
[52] Q. Shen and R. Jensen, "Selecting Informative Features with Fuzzy-Rough Sets and Its Application for Complex Systems Monitoring," *Pattern Recognition,* vol. 37, no. 7, pp. 1351-1363, 2004.
[53] A. Skowron and C. Rauszer, "The Discernibility Matrices and Functions in Information Systems," *Intelligent Decision Support,* pp. 331-362, 1992.
[54] A. Skowron and J.W. Grzymala-Busse, "From Rough Set Theory to Evidence Theory," *Advances in the Dempster-Shafer Theory of Evidence,* R. Yager, M. Fedrizzi, and J. Kasprzyk, eds. John Wiley & Sons, Inc., 1994.
[55] A. Skowron and J. Stepaniuk, "Tolerance Approximation Spaces," *Fundamenta Informaticae,* vol. 27, no. 2, pp. 245-253, 1996.
[56] A. Skowron, J. Komorowski, Z. Pawlak, and L. Polkowski, "Rough Sets Perspective on Data and Knowledge," *Handbook of Data Mining and Knowledge Discovery,* pp. 134-149, Oxford Univ. Press, 2002.
[57] A. Skowron and S.K. Pal, "Rough Sets, Pattern Recognition, and Data Mining," *Pattern Recognition Letters,* vol. 24, no. 6, pp. 829-933, 2003.
[58] D. Slezak, "Approximate Reducts in Decision Tables," *Proc. Sixth Int'l Conf., Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU '96),* pp. 1159-1164, 1996.
[59] D. Slezak, "Normalized Decision Functions and Measures for Inconsistent Decision Tables Analysis," *Fundamenta Informaticae,* vol. 44, no. 3, pp. 291-319, 2000.
[60] *Intelligent Decision Support,* R. Slowinski, ed. Kluwer Academic Publishers, 1992.
[61] R. Slowinski and D. Vanderpooten, "Similarity Relation as a Basis for Rough Approximations," *Advances in Machine Intelligence and Soft Computing,* pp. 17-33, P. Wang, ed., vol. IV, , Duke Univ. Press, 1997.
[62] J. Stefanowski and A. Tsoukiàs, "Valued Tolerance and Decision Rules," *Rough Sets and Current Trends in Computing,* pp. 212-219, 2000.
[63] R.W. Swiniarski and A. Skowron, "Rough Set Methods in Feature Selection and Recognition," *Pattern Recognition Letters,* vol. 24, no. 6, pp. 833-849, 2003.
[64] H. Thiele, "Fuzzy Rough Sets versus Rough Fuzzy Sets—An Interpretation and a Comparative Study Using Concepts of Modal Logics," Technical Report no. CI-30/98, Univ. of Dortmund, 1998.
[65] C.J. van Rijsbergen, *Information Retrieval.* London: Butterworths, 1979.
[66] J. Wang and J. Wang, "Reduction Algorithms Based on Discernibility Matrix: The Ordered Attributes Method," *J. Computer Science and Technology,* vol. 16, no. 6, pp. 489-504, 2001.
[67] J. Wróblewski, "Finding Minimal Reducts Using Genetic Algorithms," *Proc. Second Ann. Joint Conf. Information Sciences,* pp. 186-189, 1995.
[68] M. Wygralak, "Rough Sets and Fuzzy Sets—Some Remarks on Interrelations," *Fuzzy Sets and Systems,* vol. 29, pp. 241-243, 1989.
[69] Y.Y. Yao, "A Comparative Study of Fuzzy Sets and Rough Sets," *Information Sciences,* vol. 109, pp. 21-47, 1998.
[70] L.A. Zadeh, "Fuzzy Sets," *Information and Control,* vol. 8, pp. 338-353, 1965.
[71] N. Zhong, J. Dong, and S. Ohsuga, "Using Rough Sets with Heuristics for Feature Selection," *J. Intelligent Information Systems,* vol. 16, no. 3, pp. 199-214, 2001.
[72] W. Ziarko, "Variable Precision Rough Set Model," *J. Computer and System Sciences,* vol. 46, no. 1, pp. 39-59, 1993.

Richard Jensen is a PhD student in the School of Informatics at the University of Edinburgh, working in the Approximative and Qualitative Reasoning Group. His research interests include rough and fuzzy set theory, pattern recognition, information retrieval, feature selection, and swarm intelligence. He has published around 10 peer-refereed articles in these areas.

Qiang Shen is a professor with the Department of Computer Science at the University of Wales, Aberystwyth. His research interests include fuzzy and imprecise modeling, model-based inference, pattern recognition, and knowledge refinement and reuse. Dr Shen is an associate editor of the *IEEE Transactions on Fuzzy Systems* and an editorial board member of the *Fuzzy Sets and Systems Journal*. He has published around 140 peer-refereed papers in academic journals and conferences on topics within artificial intelligence and related areas.