PRIFYSGOL ABERYSTWYTH UNIVERSITY

Aberystwyth University

Extending Propositional Satisfiability to Determine Minimal Fuzzy-Rough Reducts

Tuson, Andrew; Shen, Qiang; Jensen, Richard

Publication date:
2010

Citation for published version (APA):
Tuson, A., Shen, Q., & Jensen, R. (2010). *Extending Propositional Satisfiability to Determine Minimal Fuzzy-Rough Reducts*. 1415-1422. http://hdl.handle.net/2160/4825

General rights
Copyright and moral rights for the publications made accessible in the Aberystwyth Research Portal (the Institutional Repository) are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the Aberystwyth Research Portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the Aberystwyth Research Portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

tel: +44 1970 62 2400
email: is@aber.ac.uk

Download date: 29. Mar. 2024

Extending Propositional Satisfiability to Determine Minimal Fuzzy-Rough Reducts

Richard Jensen, Andrew Tuson and Qiang Shen

***Abstract*— This paper describes a novel, principled approach to real-valued dataset reduction based on fuzzy and rough set theory. The approach is based on the formulation of fuzzy-rough discernibility matrices, that can be transformed into a satisfiability problem; an extension of rough set approaches that only apply to discrete datasets. The fuzzy-rough hybrid reduction method is then realised algorithmically by a modified version of a traditional satisifability approach. This produces an efficient and provably optimal approach to data reduction that works well on a number of machine learning benchmarks in terms of both time and classification accuracy.**

I. Introduction

There is interest in developing methodologies which are capable of dealing with imprecision and uncertainty: research currently being carried out in fuzzy and rough sets is representative of this. Many deep relationships have been established and recent studies have concluded at the complementary nature of the two methodologies. Therefore, it is desirable to extend and hybridize the underlying concepts to deal with additional aspects of data imperfection; so to offer flexibility and provide robust solutions and advanced tools for data analysis. Rough set-based feature selection is one such tool that has been shown to be highly useful at reducing data dimensionality; however, it is only directly applicable to discrete datasets. Progress has been made in terms of effective data reduction methods: work in [7] demonstrates the application of propositional satisfiability techniques to the discovery of optimal data reductions from rough set discernibility functions.

The issue of real-valued data is important and is central to real-world applications. This paper proposes a fuzzy extension to crisp discernibility matrices that is utilized for the purpose of fuzzy-rough feature selection. Additionally, the concepts in propositional satisfiability are fuzzified for use in a DPLL-like search (FRFS-SAT) to find the globally optimal subset of features. Computational results on common machine learning benchmark problems indicate that FRFS-SAT produces no reduction in classification performance compared against the original and heuristically reduced datasets. In addition, the computational requirements are not excessive, given the ability of the algorithm to guarantee optimal data reductions.

The remainder of this paper is structured as follows: in Section II, the necessary theoretical background is provided concerning the required rough set concepts. Section III introduces fuzzy discernibility matrices and how dataset reductions may be achieved in this framework. In Section IV, FRFS-SAT is detailed with corresponding algorithms and a simple walkthrough example. Experimental results that demonstrate the potential of the approach are presented in Section V. Finally, Section VI concludes the paper.

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

A. Tuson is with the Department of Computing, School of Informatics, City University, London, UK (email: andrewt@soi.city.ac.uk)

II. Theoretical Background

Rough Set Attribute Reduction (RSAR) [3] provides a filter-based tool by which knowledge may be extracted from a domain in a concise way; retaining the information content whilst reducing the amount of knowledge involved.

A. Rough Set Feature Selection

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

$$IND(P) = \{(x, y) \in \mathbb{U}^2 | \forall a \in P,\ a(x) = a(y)\} \quad (1)$$

The partition of $\mathbb{U}$, generated by *IND(P)* is denoted $\mathbb{U}$*/IND(P)* (or $\mathbb{U}$*/P* for simplicity) and can be calculated as: $\mathbb{U}/IND(P) = \otimes\{\mathbb{U}/IND(\{a\}) | a \in P\}$, where $\otimes$ is specifically defined as follows for sets A and B: $A \otimes B = \{X \cap Y | X \in A, Y \in B, X \cap Y \neq \emptyset\}$. 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 $[x]_P$.

A *decision system* $(\mathbb{U}, \mathbb{C} \cup \mathbb{D})$ is an information system in which $\mathbb{D}$ is a designated attribute or set of attributes called decision. Decision systems are often used in the context of classification. 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:

$$\underline{P}X = \{x \in \mathbb{U} \mid [x]_P \subseteq X\} \quad (2)$$

$$\overline{P}X = \{x \in \mathbb{U} \mid [x]_P \cap X \neq \emptyset\} \quad (3)$$

The tuple $\langle \underline{P}X, \overline{P}X \rangle$ is called a rough set. Let *P* and *Q* be sets of attributes inducing equivalence relations over $\mathbb{U}$, then the positive region can be defined as: $POS_P(Q) = \bigcup_{X \in \mathbb{U}/Q} \underline{P}X$. This region contains all objects of $\mathbb{U}$ that can be classified to classes of $\mathbb{U}$*/Q* using the information in attributes *P*. Using this definition of the positive region, we can define the rough set degree of

dependency of a set of attributes Q on a set of attributes P. 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

$$k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|\mathbb{U}|} \tag{4}$$

Attribute reduction is achieved by comparing equivalence relations generated by sets of attributes. Attributes are removed so that the reduced set provides the same predictive capability of the decision attribute as the original. A *reduct* R_{min} is defined as a minimal subset *R* of the initial attribute set $\mathbb{C}$ such that for a given set of attributes D, $\gamma_R(\mathbb{D})$ = $\gamma_{\mathbb{C}}(\mathbb{D})$. From the literature, R is a minimal subset if $\gamma_{R-\{a\}}(\mathbb{D}) \neq \gamma_R(\mathbb{D})$ for all $a \in R$. This means that no attributes can be removed from the subset without affecting the dependency degree. Hence, a minimal subset by this definition may not be the *global* minimum (a reduct of smallest cardinality). The intersection of all the sets in R_{all} is called the *core*, the elements of which are those attributes that cannot be eliminated without introducing more contradictions to the representation of the dataset. For many tasks a reduct of minimal cardinality (i.e. globally optimal) is ideally searched for.

B. Discernibility Matrices

Many applications of rough sets to feature selection use discernibility matrices for finding reducts. A discernibility matrix [12] of a decision table $D = (\mathbb{U}, \mathbb{C} \cup \mathbb{D})$ is a symmetric $|\mathbb{U}| \times |\mathbb{U}|$ matrix with entries defined:

$$c_{ij} = \{a \in \mathbb{C} | a(x_i) \neq a(x_j)\} \;\; i, j = 1, ..., |\mathbb{U}| \tag{5}$$

Each c_{ij} contains attributes that differ between objects i and j. To find reducts, the decision-relative discernibility matrix is of interest: this considers those object discernibilities that occur when the corresponding decision features differ. Grouping all entries containing single features forms the core dataset (features appearing in *every* reduct); they imply that at least two objects can only be distinguished by this feature alone, so must appear in all reducts.

From this, we define the discernibility function: a concise notation of how each object within the dataset may be distinguished from the others. A discernibility function f_D is a boolean function of m boolean variables $a_1^*, ..., a_m^*$ (corresponding to the attributes $a_1, ..., a_m$) defined as:

$$f_D(a_1^*, ..., a_m^*) = \wedge\{\vee c_{ij}^* | 1 \leq j \leq i \leq |\mathbb{U}|, c_{ij} \neq \emptyset\} \tag{6}$$

where $c_{ij}^* = \{a^* | a \in c_{ij}\}$. By finding the set of all prime implicants [12] of the discernibility function, all the minimal reducts of a system may be determined.

Initial work investigating the application of propositional satisfiability techniques to the discovery of crisp reducts from discernibility functions can be found in [7].

III. Fuzzy Discernibility Matrices

The RSAR process above can only operate effectively with datasets containing discrete values. There is also no way of handling noisy data. As most datasets contain real-valued attributes, it is necessary to perform a discretization step beforehand. This is typically implemented by standard fuzzification techniques, enabling linguistic labels to be associated with attribute values. However, membership degrees of attribute values to fuzzy sets are not exploited in the process of dimensionality reduction. By using *fuzzy-rough* sets [6], it is possible to use this information to better guide feature selection; this already has been shown to be a highly useful technique in reducing data dimensionality [8].

A. Fuzzy-Rough Approximations

Definitions for the fuzzy lower and upper approximations can be found in [4], [11], where a T-transitive fuzzy similarity relation is used to approximate a fuzzy concept X:

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

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

Here, I is a fuzzy implicator and T a t-norm. R_P is the fuzzy similarity relation induced by the subset of features P:

$$\mu_{R_P}(x, y) = \mathcal{T}_{a \in P}\{\mu_{R_a}(x, y)\} \tag{9}$$

$\mu_{R_a}(x, y)$ is the degree to which objects x and y are similar for feature a. Many fuzzy similarity relations can be constructed for this purpose, for example:

$$\mu_{R_a}(x, y) = \exp(-\frac{(a(x) - a(y))^2}{2{\sigma_a}^2}) \tag{10}$$

$$\mu_{R_a}(x, y) = \max(\min(\frac{(a(y) - (a(x) - \sigma_a))}{(\sigma_a)}, \frac{((a(x) + \sigma_a) - a(y))}{(\sigma_a)}), 0) \tag{11}$$

where ${\sigma_a}^2$ is the variance of feature a. As these relations do not necessarily display T-transitivity, the fuzzy transitive closure must be computed for each attribute. The combination of feature relations in equation (9) has been shown to preserve T-transitivity [15].

In a similar way to the original FRFS approach, the fuzzy positive region can be defined as:

$$\mu_{POS_{R_P}(Q)}(x) = \sup_{X \in \mathbb{U}/Q} \mu_{\underline{R_P}X}(x) \tag{12}$$

The resulting degree of dependency is:

$$\gamma'_P(Q) = \frac{\sum_{x \in \mathbb{U}} \mu_{POS_{R_P}(Q)}(x)}{|\mathbb{U}|} \tag{13}$$

A fuzzy-rough reduct R can be defined as a (locally minimal) subset of features that preserves the dependency degree of the entire dataset, i.e. $\gamma'_R(\mathbb{D}) = \gamma'_{\mathbb{C}}(\mathbb{D})$. Core features may be determined by considering the change in dependency of the full set of conditional features when individual attributes are removed:

$$Core(\mathbb{C}) = \{a \in \mathbb{C} | \gamma'_{\mathbb{C}-\{a\}}(Q) < \gamma'_{\mathbb{C}}(Q)\} \tag{14}$$

B. Fuzzy Discernibility Matrix-based FS

There are two main branches of research in crisp rough set-based FS: those based on the dependency degree and those based on discernibility matrices. The developments above are solely concerned with the extension of the dependency degree to the fuzzy-rough case. Hence, methods constructed based on the crisp dependency degree can be employed for fuzzy-rough FS. By extending the discernibility matrix to the fuzzy case, it is possible to employ approaches similar to those in crisp rough set FS to determine fuzzy-rough reducts. A first step toward this is presented in [14] where a crisp discernibility matrix is constructed for fuzzy-rough selection. A threshold is used, breaking the rough set ideology, which determines which features are to appear in the matrix entries. However, information is lost as membership degrees are not considered. Search based on the crisp discernibility may result in reducts that are not true fuzzy-rough reducts.

1) Fuzzy Discernibility: We extend the crisp discernibility matrix by employing fuzzy clauses. Entries in the fuzzy discernibility matrix is a fuzzy set, to which every feature belongs to a certain degree. The extent to which a feature a belongs to the fuzzy clause C_{ij} is determined by the fuzzy discernibility measure:

$$\mu_{C_{ij}}(a) = N(\mu_{R_a}(i,j)) \tag{15}$$

where N denotes fuzzy negation and $\mu_{R_a}(i,j)$ is the fuzzy similarity of objects i and j, and hence $\mu_{C_{ij}}(a)$ is a measure of the fuzzy discernibility. For the crisp case, if $\mu_{C_{ij}}(a)$ = 1 then the two objects are distinct for this feature; if $\mu_{C_{ij}}(a)$ = 0, the two objects are identical. For fuzzy cases where $\mu_{C_{ij}}(a) \in (0,1)$, the objects are partly discernible. (The choice of fuzzy similarity relation must be identical to that of the fuzzy-rough dependency degree approach to find corresponding reducts.) Each entry in the fuzzy indiscernibility matrix is a set of attributes and their memberships:

$$C_{ij} = \{a_x | a \in \mathbb{C}, x = N(\mu_{R_a}(i,j))\}\ i,j = 1,...,|\mathbb{U}| \tag{16}$$

For example, an entry C_{ij} in the fuzzy discernibility matrix might be: $\{a_{0.4}, b_{0.8}, c_{0.2}, d_{0.0}\}$. This denotes that $\mu_{C_{ij}}(a) = 0.4$, $\mu_{C_{ij}}(b) = 0.8$, etc. In crisp discernibility matrices, these values are either 0 or 1 as the underlying relation is an equivalence relation. The example clause can be viewed as indicating the value of each feature - the extent to which the feature discriminates between the two objects i and j. The core of the dataset is defined as:

$$\begin{aligned} Core(\mathbb{C}) = \{a \in \mathbb{C} | \exists C_{ij}, \mu_{C_{ij}}(a) > 0, \\ \forall f \in \{\mathbb{C} - a\}\ \mu_{C_{ij}}(f) = 0\} \end{aligned} \tag{17}$$

2) Fuzzy Discernibility Function: As with the crisp approach, the entries in the matrix can be used to construct the fuzzy discernibility function:

$$f_D(a_1^*, ..., a_m^*) = \wedge\{\vee C_{ij}^* | 1 \leq j < i \leq |\mathbb{U}|\} \tag{18}$$

where $C_{ij}^* = \{a_x^* | a_x \in C_{ij}\}$. The function returns values in $[0,1]$, which can be seen to be a measure of the extent to which the function is satisfied for a given assignment of truth values to variables. To discover reducts from the fuzzy discernibility function, the task is to find the minimal assignment of the value 1 to the variables such that the formula is maximally satisfied. By setting all variables to 1, the maximal value for the function can be obtained as this provides the most discernibility between objects.

3) Decision-relative Fuzzy Discernibility Matrix: As with the crisp discernibility matrix, for a decision system the decision feature must be taken into account for achieving reductions; only those clauses with different decision values are included in the crisp discernibility matrix. For the fuzzy version, this is encoded as:

$$\begin{aligned} f_D(a_1^*, ..., a_m^*) = \{\wedge\{\{\vee C_{ij}^*\} \leftarrow q_{N(\mu_{R_q}(i,j))}\} | \\ 1 \leq j < i \leq |\mathbb{U}|\} \end{aligned} \tag{19}$$

for decision feature q, where $\leftarrow$ denotes fuzzy implication. This allows the extent to which decision values differ to affect the overall satisfiability of the clause. If $\mu_{C_{ij}}(q)$ = 1 then this clause provides maximum discernibility (i.e. the two objects are maximally different according to the fuzzy similarity measure). When the decision is crisp and crisp equivalence is used, $\mu_{C_{ij}}(q)$ becomes 0 or 1.

IV. FRFS-SAT

Reducts are calculated via the fuzzy clauses from by the construction of the fuzzy discernibility function above. Crisp discernibility matrices can be adapted with suitable extensions. The aim here is to determine those reducts that are minimal in the global sense (i.e. of smallest cardinality). Thus, heuristic techniques are not applicable as the resulting reducts may not satisfy this property, and there is no computationally efficient way of determining this for a particular reduct. This section proposes a fuzzy extension to propositional satisfiability for the purpose of determining globally minimal reducts.

A. Formulation

The degree of satisfaction of a clause C_{ij} for a subset of features P is defined as:

$$SAT_P(C_{ij}) = \mathcal{S}_{a \in P}\{\mu_{C_{ij}}(a)\} \tag{20}$$

for a t-conorm $\mathcal{S}$. Returning to the example clause $\{a_{0.4}, b_{0.8}, c_{0.2}, d_{0.0}\}$, if the subset $P = \{a, c\}$ is chosen, the resulting degree of satisfaction of the clause is

$$SAT_P(C_{ij}) = \mathcal{S}\{0.4, 0.2\} = 0.6$$

using the Łukasiewicz t-conorm, $\min(1, x + y)$.

In traditional (crisp) propositional satisfiability, a clause is fully satisfied if at least one variable in the clause has been set to `true`. For the fuzzy case, clauses may be satisfied to a certain degree depending on which variables have been assigned the value `true`. By setting $P = \mathbb{C}$, the maximum satisfiability degree of a clause may be obtained:

$$maxSAT_{ij} = SAT_{\mathbb{C}}(C_{ij}) = \mathcal{S}_{a \in \mathbb{C}}\{\mu_{C_{ij}}(a)\} \tag{21}$$

This is the maximal amount that clause C_{ij} can be satisfied. The maximum satisfiability degree of the example clause is $\mathcal{S}(0.4, 0.8, 0.2, 0.0)$ which evaluates to 1 if the Łukasiewicz t-conorm is used. Here it can be seen that, depending on the t-conorm used, clauses may in fact be maximally satisfied by the selection of several sub-maximal features. Using the $\max$ t-conorm, the maximum satisfiability degree is 0.8, obtained only by the inclusion of feature b in P.

In this setting, a fuzzy-rough reduct corresponds to a (minimal) truth assignment to variables such that each clause has been satisfied to its maximum extent. See the appendix for a proof that fuzzy-rough reducts maximally satisfy the set of clauses for a given dataset.

B. Algorithm

The DPLL-based algorithm for finding minimal subsets is in figure 1, where search is conducted in a depth-first manner. The key operation in this procedure is the unit propagation step, unitPropagate(CL), in lines (6) and (7). Clauses in the formula that contain a single literal will only be satisfied if that literal is assigned the value `true` (unit clauses). Unit propagation examines the current formula for unit clauses and assigns the appropriate value to the literal they contain. The elimination of a literal can create new unit clauses, and thus unit propagation eliminates variables by repeated passes until there is no unit clause in the formula. The order of the unit clauses within the formula makes no difference to the results or the efficiency of the process.

Branching occurs at lines (10) to (14) via the function selectLiteral(CL). Here, the next literal is chosen heuristically from the current formula, assigned the value `true`, and the search continues. If this branch eventually results in unsatisfiability, the procedure assigns the value `false` to this literal instead and continue the search. Choosing good branching literals is important - different branching heuristics may produce drastically different sized search trees for the same basic algorithm, affecting the efficiency of the solver.

One heuristic is to select the variable whose fuzzy discernibility is non-zero in the most clauses of the current set of clauses. Alternatively, the sum of the fuzzy discernibilities for a particular attribute across all clauses gives a good indication of attribute importance. This is the heuristic adopted.

Some pruning takes place in the search by remembering the size of the currently considered subset d and the smallest optimal subset encountered so far D. If the number of variables currently assigned the value `true` equals the number of those in the presently optimal subset then any further search down this branch will not result in a smaller optimal subset. Also, if an empty clause is generated during UPDATE-FALSE, the algorithm stops the search down this branch.

Line (3) is reached when all clauses have been maximally satisfied (a fuzzy-rough reduct has been reached) and the corresponding variable assignment is outputted. The final outputted variable assignment is the globally minimal reduct.

Figure 2 shows the update of the current clause list if the variable x is set to `true`. The updated clause list is stored in CL' and returned upon completion. Line (4) determines

DPLL-SOLVE(d, CL, D).
d, the current depth of search;
CL, the current list of clauses;
D, the depth of the best reduct found so far (initially $|\mathbb{C}|$).

```
(1)  if (d ≥ D) or (CL == null)
(2)      // Further search down this branch is unnecessary
(3)  else if (CL.size() == 0) and (d < D)
(4)      D ← d
(5)      output current assignment
(6)  else if (CL contains a unit clause {l})
(7)      CL' ← unitPropagate(CL)
(8)      DPLL-SOLVE(d + 1,CL',D)
(9)  else
(10)     x ← selectLiteral(CL)
(11)     CL' ← UPDATE-TRUE(CL, x)
(12)     DPLL-SOLVE(d + 1,CL',D)
(13)     CL' ← UPDATE-FALSE(CL, x)
(14)     DPLL-SOLVE(d,CL',D)
```

Fig. 1. The DPLL-SOLVE algorithm

UPDATE-TRUE(CL, x).
CL, the current clause list;
x, the variable to be set to `true`.

```
(1)  CL' ← ∅
(2)  foreach C ∈ CL
(3)      if (!isSatisfied(C))
(4)          CL' ← CL' ∪ C
(5)  return CL'
```

Fig. 2. The UPDATE-TRUE algorithm

if the clause C will be maximally satisfied if variable x is set to true. If not, the fuzzy clause is retained and added to the updated clause list. Once a clause is maximally satisfied, it is not considered further down this branch in the search.

When the chosen literal is assigned the value `false` (i.e. does not appear in subsets beyond this branching point), the fuzzy clauses are updated according to Figure 3. Each clause C in the current set of clauses is examined. In line (4), $|C|$ denotes the number of literals in the clause that can be set to `true`; if this is zero, then this clause cannot be satisfied. Line (4) also checks to see if the clause is satisfiable, i.e. could potentially reach the maximum satisfiability degree if further literals are chosen. If not, the current variable assignment cannot lead to a fuzzy-rough reduct, and so search down this branch need not be considered.

1) Example: Table I illustrates the operation of FRFS-SAT, using an example dataset. The fuzzy connectives used are the Łukasiewicz t-norm ($\max(x + y - 1, 0)$) and the Łukasiewicz fuzzy implicator ($\min(1 - x + y, 1)$). As recommended in [4], the Łukasiewicz t-norm is used as this

UPDATE-FALSE(CL, x).
CL, the current clause list;
x, the variable to be set to `false`.

(1) $CL' \leftarrow \emptyset$
(2) **foreach** $C \in CL$
(3) **if** ($|C|$==0) **or** (!isSatisfiable(C))
(4) **return** null //Further search is pointless
(5) **else** $CL' \leftarrow CL' \cup C$
(6) **return** CL'

Fig. 3. The UPDATE-FALSE algorithm

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

TABLE I
EXAMPLE DATASET

produces fuzzy T-equivalence relations dual to that of a pseudo-metric. The use of the Łukasiewicz fuzzy implicator is also recommended as it is both a residual and S-implicator.

Using the fuzzy similarity measure in (11), the resulting relations are as follows for each feature in the dataset:

$$R_a(x,y) \begin{pmatrix} 1.0 & 1.0 & 0.699 & 0.0 & 0.0 & 0.0 \\ 1.0 & 1.0 & 0.699 & 0.0 & 0.0 & 0.0 \\ 0.699 & 0.699 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.699 & 0.699 \\ 0.0 & 0.0 & 0.0 & 0.699 & 1.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 0.699 & 1.0 & 1.0 \end{pmatrix}$$

$$R_b(x,y) \begin{pmatrix} 1.0 & 0.0 & 0.568 & 1.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.137 \\ 0.568 & 0.0 & 1.0 & 0.568 & 0.568 & 0.0 \\ 1.0 & 0.0 & 0.568 & 1.0 & 1.0 & 0.0 \\ 1.0 & 0.0 & 0.568 & 1.0 & 1.0 & 0.0 \\ 0.0 & 0.137 & 0.0 & 0.0 & 0.0 & 1.0 \end{pmatrix}$$

$$R_c(x,y) \begin{pmatrix} 1.0 & 0.0 & 0.036 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.036 & 0.518 & 0.518 & 0.518 \\ 0.036 & 0.036 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.518 & 0.0 & 1.0 & 1.0 & 1.0 \\ 0.0 & 0.518 & 0.0 & 1.0 & 1.0 & 1.0 \\ 0.0 & 0.518 & 0.0 & 1.0 & 1.0 & 1.0 \end{pmatrix}$$

Next, the fuzzy discernibility matrix needs to be constructed based on the fuzzy discernibility given in equation (15). For objects 2 and 3, the resulting fuzzy clause is $\{a_{0.301} \vee b_{1.0} \vee c_{0.964}\} \leftarrow q_{1.0}$,

The fuzzy discernibility of objects 2 and 3 for attribute a is 0.301, indicating that the objects are partly discernible for this feature. The objects are fully discernible with respect to the decision feature, indicated by $q_{1.0}$. The set of clauses is:

$$\begin{aligned}
C_{12}: &\quad \{a_{0.0} \vee b_{1.0} \vee c_{1.0}\} \leftarrow q_{1.0} \\
C_{13}: &\quad \{a_{0.301} \vee b_{0.432} \vee c_{0.964}\} \leftarrow q_{0.0} \\
C_{14}: &\quad \{a_{1.0} \vee b_{0.0} \vee c_{1.0}\} \leftarrow q_{1.0} \\
C_{15}: &\quad \{a_{1.0} \vee b_{0.0} \vee c_{1.0}\} \leftarrow q_{1.0} \\
C_{16}: &\quad \{a_{1.0} \vee b_{1.0} \vee c_{1.0}\} \leftarrow q_{0.0} \\
C_{23}: &\quad \{a_{0.301} \vee b_{1.0} \vee c_{0.964}\} \leftarrow q_{1.0} \\
C_{24}: &\quad \{a_{1.0} \vee b_{1.0} \vee c_{0.482}\} \leftarrow q_{0.0} \\
C_{25}: &\quad \{a_{1.0} \vee b_{1.0} \vee c_{0.482}\} \leftarrow q_{0.0} \\
C_{26}: &\quad \{a_{1.0} \vee b_{0.863} \vee c_{0.482}\} \leftarrow q_{1.0} \\
C_{34}: &\quad \{a_{1.0} \vee b_{0.431} \vee c_{1.0}\} \leftarrow q_{1.0} \\
C_{35}: &\quad \{a_{1.0} \vee b_{0.431} \vee c_{1.0}\} \leftarrow q_{1.0} \\
C_{36}: &\quad \{a_{1.0} \vee b_{1.0} \vee c_{1.0}\} \leftarrow q_{0.0} \\
C_{45}: &\quad \{a_{0.301} \vee b_{0.0} \vee c_{0.0}\} \leftarrow q_{0.0} \\
C_{46}: &\quad \{a_{0.301} \vee b_{1.0} \vee c_{0.0}\} \leftarrow q_{1.0} \\
C_{56}: &\quad \{a_{0.0} \vee b_{1.0} \vee c_{0.0}\} \leftarrow q_{1.0}
\end{aligned}$$

Due to the properties of implicators, all clauses with $q_{0.0}$ may be removed without influencing the final outputted reduct, hence the clause list can be reduced to (with duplicates removed):

$$\begin{aligned}
C_{12}: &\quad \{a_{0.0} \vee b_{1.0} \vee c_{1.0}\} \leftarrow q_{1.0} \\
C_{14}: &\quad \{a_{1.0} \vee b_{0.0} \vee c_{1.0}\} \leftarrow q_{1.0} \\
C_{23}: &\quad \{a_{0.301} \vee b_{1.0} \vee c_{0.964}\} \leftarrow q_{1.0} \\
C_{26}: &\quad \{a_{1.0} \vee b_{0.863} \vee c_{0.482}\} \leftarrow q_{1.0} \\
C_{34}: &\quad \{a_{1.0} \vee b_{0.431} \vee c_{1.0}\} \leftarrow q_{1.0} \\
C_{46}: &\quad \{a_{0.301} \vee b_{1.0} \vee c_{0.0}\} \leftarrow q_{1.0} \\
C_{56}: &\quad \{a_{0.0} \vee b_{1.0} \vee c_{0.0}\} \leftarrow q_{1.0}
\end{aligned}$$

The DPLL-SOLVE algorithm is then used to determine the minimal reduct. Clause C_{56} is a unit clause (here feature b is a core attribute), so variable b is set to `true`. The UPDATE-TRUE procedure is then executed, removing all clauses that are now maximally satisfied as a result of this assignment:

$$\begin{aligned}
C_{14}: &\quad \{a_{1.0} \vee 0.0 \vee c_{1.0}\} \leftarrow q_{1.0} \\
C_{26}: &\quad \{a_{1.0} \vee 0.863 \vee c_{0.482}\} \leftarrow q_{1.0} \\
C_{34}: &\quad \{a_{1.0} \vee 0.431 \vee c_{1.0}\} \leftarrow q_{1.0}
\end{aligned}$$

Next, line (12) of the algorithm is executed. There are no unit clauses, so line (10) is reached and the variable a is chosen as the sum of its fuzzy discernibilities is greater than that of c. With a set to `true`, all clauses have been maximally satisfied and $\{a, b\}$ is outputted. The algorithm terminates at this point, as the choice of setting b to `false` is unavailable as b was chosen via a unit clause (and hence must be set to `true`).

C. Simplification

Crisp discernibility matrices are simplified by removing duplicate entries and clauses that are supersets of others. This can be achieved for fuzzy discernibility matrices: duplicate clauses can be removed as a subset that satisfies one clause to a certain degree will always satisfy the other to the same degree. Also, clauses whose decision component is zero can also be removed due to the properties of fuzzy implication.

A further degree of simplification is obtained by an extension of the crisp approach where clauses that are supersets of others are removed (termed absorption), for the fuzzy case:

$$S(C_{ij}, C_{kl}) = \frac{\sum_{a \in \mathbb{C}} \mathcal{T}(\mu_{C_{ij}}(a), \mu_{C_{kl}}(a))}{\sum_{a \in \mathbb{C}} \mu_{C_{ij}}(a)} \quad (22)$$

If $S(C_{ij}, C_{kl}) = 1$ then clause C_{kl} is subsumed by clause C_{ij} and can be removed. Of course, further simplification techniques from the literature on crisp discernibility matrices and functions could be extended and applied, but only fuzzy absorption is considered here.

Returning to the example, the original set of clauses used as input to DPLL-SOLVE are:

$$\begin{array}{lrcl}
C_{12}: & \{a_{0.0} \vee b_{1.0} \vee c_{1.0}\} & \leftarrow & q_{1.0} \\
C_{14}: & \{a_{1.0} \vee b_{0.0} \vee c_{1.0}\} & \leftarrow & q_{1.0} \\
C_{23}: & \{a_{0.301} \vee b_{1.0} \vee c_{0.964}\} & \leftarrow & q_{1.0} \\
C_{26}: & \{a_{1.0} \vee b_{0.863} \vee c_{0.482}\} & \leftarrow & q_{1.0} \\
C_{34}: & \{a_{1.0} \vee b_{0.431} \vee c_{1.0}\} & \leftarrow & q_{1.0} \\
C_{46}: & \{a_{0.301} \vee b_{1.0} \vee c_{0.0}\} & \leftarrow & q_{1.0} \\
C_{56}: & \{a_{0.0} \vee b_{1.0} \vee c_{0.0}\} & \leftarrow & q_{1.0}
\end{array}$$

The fuzzy absorption simplification process compares each pair of clauses and removes those that are subsumed. For example, clauses C_{46} and C_{23}:

$$\begin{aligned}
S(C_{46}, C_{23}) &= \frac{\sum_{a \in \mathbb{C}} \mathcal{T}(\mu_{C_{46}}(a), \mu_{C_{23}}(a))}{\sum_{a \in \mathbb{C}} \mu_{C_{46}}(a)} \\
&= \frac{\mathcal{T}(0.301, 0.301) + \mathcal{T}(1, 1) + \mathcal{T}(0, 0.964)}{1.301}
\end{aligned}$$

In this case, $S(C_{46}, C_{23}) = 1$ so clause C_{23} can be removed. Any assignment of truth values to variables such that C_{46} is maximally satisfied also implies that C_{23} is maximally satisfied. The reverse is not true, so C_{23} provides no further information than that already possessed by C_{46}. Applying this process to all clauses results in:

$$\begin{array}{lrcl}
C_{14}: & \{a_{1.0} \vee b_{0.0} \vee c_{1.0}\} & \leftarrow & q_{1.0} \\
C_{26}: & \{a_{1.0} \vee b_{0.863} \vee c_{0.482}\} & \leftarrow & q_{1.0} \\
C_{56}: & \{a_{0.0} \vee b_{1.0} \vee c_{0.0}\} & \leftarrow & q_{1.0}
\end{array}$$

The number of clauses has been reduced to 3 from the original 7, and DPLL search from this point is straightforward resulting in the reduct $\{a, b\}$. The subset $\{b, c\}$ is also a reduct, as discovered by the original FRFS algorithm [8]. Again, use of the Łukasiewicz t-conorm can result in a clause being maximally satisfied with the choice of several submaximal features. In this case, $\mathcal{S}(0.863, 0.482) = 1$, so $\{b, c\}$ is a valid fuzzy-rough reduct.

This simplification process is effective, but computationally expensive: the process must compare each clause with every other clause in the clause list. For the worst case, $c = (n^2 - n)/2$ clauses are generated initially, so $(c^2 - c)/2$ clause comparisons are made. This can be reduced by integrating the simplification into the discernibility matrix construction process; as clauses are generated, they are checked for fuzzy absorption against existing clauses and vice versa.

Another simplification method for crisp discernibility matrices is local strong compressibility [13]. If a subset of attributes is simultaneously present or absent in the set of clauses, then they can be replaced by a single representative attribute (since all attributes in this class possess exactly the same information, then with one of the attributes selected, the rest are redundant). Figure 4 shows the extension of this concept to the fuzzy case, where attribute a_1 is tested to see if it is redundant in the presence of attribute a_2.

FUZZY-COMPRESSIBILITY(CL, a_1, a_2).
CL, the current clause list;
a_1,a_2, conditional attributes.

```
(1) foreach C ∈ CL
(2)     if (S(μ_C(a_1), μ_C(a_2)) > μ_C(a_2))
(3)         return false
(4) return true
```

Fig. 4. The FUZZY-COMPRESSIBILITY algorithm

D. Unsupervised selection

The use of rough and fuzzy-rough sets for unsupervised feature selection has been investigated [10]. This is achieved in this framework by setting all decision components to 1, specifying that all pairs of objects must be distinguishable.

V. EXPERIMENTATION

This section presents the initial experimental evaluation of the proposed method on 9 benchmark datasets from [2] and [9]. The number of conditional features ranges from 10 to 39 over the datasets. The methods used in the comparison were the fuzzy dependency, fuzzy boundary region and fuzzy discernibility [8] measures, all using a greedy hill-climbing search process. Additionally, two alternative search methods were used with the fuzzy dependency measure, genetic algorithms (GA) and particle swarm optimization (PSO), in order to search for the smallest subsets[1].

JRip [5] was employed for the purpose of evaluating the resulting subsets. JRip learns propositional rules by repeatedly growing rules and pruning them. During the growth phase, features are added greedily until a termination condition is satisfied. Features are then pruned in the next phase subject to a pruning metric. Once the ruleset is generated, a further optimization is performed where classification rules are evaluated and deleted based on their performance on randomized data.

For the experiments themselves, 10×10-fold cross validation was performed, where each feature selection algorithm is applied to the training folds and then the resulting subsets used to reduce the test fold each time. The average subset size found for each method can be seen in table II and the

[1]All evaluation measures described in this paper have been implemented in Weka [16]. The program can be downloaded from http://users.aber.ac.uk/rkj/book/programs.php

TABLE II
NUMBER OF FEATURES SELECTED

Dataset	Unreduced	FRFS-SAT	Depend.	Boundary	Discern.	GA	PSO
Australian	15	12.70	12.85	12.85	12.85	12.77	12.70
Cleveland	14	7.54	7.62	7.65	7.62	**8.10**	7.80
Glass	14	8.44	**9.00**	8.44	8.44	8.44	8.44
Heart	10	7.00	7.07	7.07	7.13	**7.52**	7.12
Ionosphere	35	5.99	**6.99**	**6.99**	**7.04**	**9.61**	**7.33**
Olitos	26	4.98	5.00	4.99	5.00	**6.03**	5.08
Water 2	39	5.85	5.99	5.99	5.99	**7.00**	**6.64**
Water 3	39	5.87	6.00	6.00	5.99	**7.42**	**6.80**
Wine	14	4.51	**5.00**	**4.87**	4.82	**5.01**	**4.98**

corresponding average classification accuracies can be found in table III. Numbers in bold indicate a statistically worse performance when compared to FRFS-SAT.

From this, FRFS-SAT finds the globally optimal reduct for each dataset without a statistically significant loss in classification accuracy. The three measures that employ a hill-climbing search strategy all locate reducts of a small size, though not necessarily globally optimal. The boundary region measure and discernibility measure appear to be more informed heuristics. The difficulty of finding globally minimal reducts can be seen in the results for the more advanced search strategies (GA and PSO). Neither method consistently finds such reducts: PSO always finds the global minimum for two datasets (Australian and Glass), the GA approach only finds the minimum for the Glass dataset. Overall the PSO method outperforms the GA approach. However, the reducts found by these methods are not guaranteed to be minimal.

The average time taken by the algorithms when performing selection can be found in table IV. The timings for FRFS-SAT include the time taken to calculate the fuzzy discernibility matrix as well as the search itself. It can be seen that, in general, FRFS-SAT can find globally optimal reducts in a similar amount of time to the other methods. However, as the dimensionality increases an increasing amount of time is spent verifying that the discovered reduct is indeed globally optimal, which is the case for the Water datasets.

VI. CONCLUSIONS

This paper has presented an extension of the discernibility matrix to the fuzzy case, allowing features to belong to entries to a certain degree. Based on this, the propositional satisfiability problem has been extended to allow SAT-style search of the resulting fuzzy clauses. From these, the globally minimal reduct for a dataset can be calculated.

Further work in this area will include experimental investigation of the proposed method and the impact of the choice of relations and connectives. Additionally, the development of fuzzy discernibility matrices here allows the extension of many existing crisp techniques for the purposes of finding fuzzy-rough reducts. In particular, other SAT solution techniques may be applied that should be able to discover such subsets, guaranteeing their minimality. The performance may also be improved through simplifying the fuzzy discernibility function further. This could be achieved by considering the properties of the fuzzy connectives and removing clauses that are redundant in the presence of others.

APPENDIX

Theorem 1: FRFS-SAT reducts are fuzzy-rough reducts. Suppose that $P \subseteq \mathbb{C}$, a is an arbitrary conditional feature that belongs to the dataset and q is the decision feature. If P maximally satisfies the fuzzy discernibility function then P is a fuzzy-rough reduct.

Proof: The fuzzy positive region for a subset P is

$$\mu_{POS_{R_P}(Q)}(x) = \sup_{X \in \mathbb{U}/Q} \inf_{y \in \mathbb{U}} \{\mu_{R_P}(x,y) \rightarrow \mu_X(y)\}$$

The dependency function is maximized when each x belongs maximally to the fuzzy positive region. Hence,

$$\inf_{x \in \mathbb{U}} \sup_{X \in \mathbb{U}/Q} \inf_{y \in \mathbb{U}} \{\mu_{R_P}(x,y) \rightarrow \mu_X(y)\}$$

is maximized only when P is a fuzzy-rough reduct. This can be rewritten as the following:

$$\inf_{x,y \in \mathbb{U}} \{\mu_{R_P}(x,y) \rightarrow \mu_{R_q}(x,y)\}$$

when using a fuzzy similarity relation in the place of crisp decision concepts, as $\mu_{[x]_R} = \mu_R(x,y)$ [6]. Each $\mu_{R_P}(x,y)$ is constructed from the t-norm of its constituent relations:

$$\inf_{x,y \in \mathbb{U}} \{T_{a \in P}(\mu_{R_a}(x,y)) \rightarrow \mu_{R_q}(x,y)\}$$

This may be reformulated as

$$\inf_{x,y \in \mathbb{U}} \{S_{a \in P}(\mu_{R_a}(x,y) \rightarrow \mu_{R_q}(x,y))\} \tag{23}$$

Considering the fuzzy discernibility matrix approach, the fuzzy discernibility function is maximally satisfied when

$$\{\wedge\{\{\vee C^*_{xy}\} \leftarrow q_{N(\mu_{R_q}(x,y))}\} | 1 \leq y < x \leq |\mathbb{U}|\}$$

is maximized. This can be rewritten as:

$$T_{x,y \in \mathbb{U}}(S_{a \in P}(N(\mu_{R_a}(x,y))) \leftarrow N(\mu_{R_q}(x,y)))$$

because each clause C_{xy} is generated by considering the fuzzy similarity of values of each pair of objects x, y.

TABLE III

JRIP CLASSIFICATION ACCURACIES (%)

Dataset	Unreduced	FRFS-SAT	Depend.	Boundary	Discern.	GA	PSO
Australian	85.36	85.00	85.23	85.23	85.32	84.75	84.13
Cleveland	54.16	53.93	54.03	53.96	54.09	53.96	54.60
Glass	67.05	65.34	67.05	65.34	65.34	65.34	65.34
Heart	79.19	76.30	75.78	75.78	75.41	76.33	75.44
Ionosphere	87.09	86.35	87.13	87.13	84.78	83.30	86.48
Olitos	68.83	61.67	62.75	64.00	62.08	59.67	61.92
Water 2	82.64	81.87	83.56	83.56	81.87	83.13	80.13
Water 3	82.44	81.41	81.51	81.51	82.08	81.33	76.46
Wine	93.18	90.29	91.96	91.62	89.53	89.09	89.74

TABLE IV

TIME TAKEN FOR FEATURE SELECTION (S)

Dataset	FRFS-SAT	Depend.	Boundary	Discern.	GA	PSO
Australian	4.21	7.24	20.07	2.90	12.52	34.07
Cleveland	0.83	0.97	3.25	0.53	2.68	6.63
Glass	0.47	0.34	1.23	0.20	0.68	2.17
Heart	0.60	0.78	1.62	0.35	2.19	5.74
Ionosphere	19.88	1.88	3.51	0.78	2.00	9.20
Olitos	2.41	0.26	0.75	0.14	0.11	1.48
Water 2	97.72	4.86	11.24	1.50	0.92	19.62
Water 3	116.86	4.87	13.50	1.72	2.36	19.69
Wine	0.70	0.27	0.66	0.13	0.75	1.83

Through the properties of the fuzzy connectives, this may be rewritten as:

$$T_{x,y\in\mathbb{U}}(S_{a\in P}(\mu_{R_a}(x,y) \rightarrow \mu_{R_q}(x,y))) \qquad (24)$$

When this is maximized, (23) is maximized and so the subset P must be a fuzzy-rough reduct.

■

REFERENCES

[1] R.B. Bhatt and M. Gopal, "On the compact computational domain of fuzzy-rough sets," *Pattern Recognition Letters*, vol. 26, no. 11, pp. 1632–1640, 2005.

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

[4] M. De Cock, C. Cornelis, and E.E. Kerre, "Fuzzy Rough Sets: The Forgotten Step," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 1, pp. 121–130, 2007.

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

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

[7] R. Jensen, Q. Shen and A. Tuson, "Finding Rough Set Reducts with SAT," Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, LNAI 3641, pp. 194–203, 2005.

[8] R. Jensen and Q. Shen, "New approaches to Fuzzy-Rough Feature Selection", *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 824–838, 2009.

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

[10] N. Mac Parthalain and R. Jensen, "Measures for Unsupervised Fuzzy-Rough Feature Selection," To appear in: Proceedings of the International Conference on Intelligent Systems Design and Applications (ISDA'09).

[11] A.M. Radzikowska and E.E. Kerre, "A comparative study of fuzzy rough sets," *Fuzzy Sets and Systems*, vol. 126, no. 2, pp. 137–155, 2002.

[12] A. Skowron and C. Rauszer, "The discernibility matrices and functions in Information Systems," In: *Intelligent Decision Support*, Kluwer Academic Publishers, pp. 331-362, 1992.

[13] J.A. Starzyk, D.E. Nelson, and K. Sturtz, "A Mathematical Foundation for Improved Reduct Generation in Information Systems," *Knowl. Inf. Syst.* 2(2): 131–146, 2000.

[14] G.C.Y. Tsang, D. Chen, E.C.C. Tsang, J.W.T. Lee, and D.S. Yeung, "On attributes reduction with fuzzy rough sets," *Proc. 2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2775–2780, 2005.

[15] M. Wallace, Y. Avrithis and S. Kollias, "Computationally efficient supt transitive closure for sparse fuzzy binary relations," *Fuzzy Sets and Systems*, vol. 157, no. 3, pp. 341–372, 2006.

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