Available online at www.sciencedirect.com
SCIENCE DIRECT®
Fuzzy Sets and Systems 149 (2005) 5–20
FUZZY sets and systems
www.elsevier.com/locate/fss

Fuzzy-rough data reduction with ant colony optimization

Richard Jensen*, Qiang Shen

Department of Computer Science, The University of Wales, Aberystwyth, Penglais, Aberystwyth, Ceredigion, Wales, UK

Available online 21 August 2004

Abstract

Feature selection 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. In particular, solution to this has found successful application in tasks that involve datasets 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. Rough set theory has been used as such a dataset pre-processor with much success, but current methods are inadequate at finding *minimal* reductions, the smallest sets of features possible. To alleviate this difficulty, a feature selection technique that employs a hybrid variant of rough sets, fuzzy-rough sets, has been developed recently and has been shown to be effective. However, this method is still not able to find the optimal subsets regularly. This paper proposes a new feature selection mechanism based on ant colony optimization in an attempt to combat this. The method is then applied to the problem of finding optimal feature subsets in the fuzzy-rough data reduction process. The present work is applied to complex systems monitoring and experimentally compared with the original fuzzy-rough method, an entropy-based feature selector, and a transformation-based reduction method, PCA. Comparisons with the use of a support vector classifier are also included.

Keywords: Data reduction; Fuzzy-rough sets; Ant colony optimization; Feature selection

1. Introduction

The main aim of feature selection (FS) is to determine a minimal feature subset from a problem domain while retaining a suitably high accuracy in representing the original features [6]. In real world problems

* Corresponding author.
E-mail addresses: rkj@aber.ac.uk (R. Jensen), qqs@aber.ac.uk (Q. Shen).

0165-0114/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2004.07.014

FS is a must due to the abundance of noisy, irrelevant or misleading features. For instance, by removing these factors, learning from data techniques such as text processing and web content classification can benefit greatly. Given a feature set size n, the task of FS can be seen as a search for an "optimal" feature subset through the competing 2^n candidate subsets. The definition of what an optimal subset is may vary, depending on the problem to be solved. Although an exhaustive method may be used for this purpose, it is quite impractical for most datasets. Usually FS algorithms involve heuristic or random search strategies in an attempt to avoid this prohibitive complexity. However, the degree of optimality of the final feature subset is often reduced.

Rough set theory (RST) [18] has been used successfully as a selection tool to discover data dependencies and reduce the number of attributes contained in a dataset by purely structural methods [5,12]. Given a dataset with discretized attribute values, by the use of rough sets it is possible to find a subset (termed a *reduct*) of the original attributes using rough sets that are the most informative; all other attributes can be removed from the dataset with minimal information loss. 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 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 the standard RST-based approach they are considered to be as different as two values of a different order of magnitude. Dataset discretization must take place before reduction methods based on crisp rough sets can be applied. This is often still inadequate, however, as the degrees of membership of values to discretized values are not considered at all.

In order to combat this, a data reduction method based on *fuzzy-rough* sets has been developed [14]. Fuzzy-rough sets encapsulate the related but distinct concepts of vagueness (for fuzzy sets [23]) and indiscernibility (for rough sets), both of which occur as a result of uncertainty in knowledge [10]. The fuzzy-rough set-based approach considers the extent to which fuzzified values are similar. Previously, an incremental hill-climbing algorithm was employed to discover the best feature subset. However, this often led to the discovery of non-optimal feature subsets, both in terms of the resulting dependency measure and the subset size.

Swarm intelligence (SI) is the property of a system whereby the collective behaviours of simple agents interacting locally with their environment cause coherent functional global patterns to emerge [3]. SI provides a basis with which it is possible to explore collective (or distributed) problem solving without centralized control or the provision of a global model. For example, ants are capable of finding the shortest route between a food source and their nest without the use of visual information and hence possess no global world model, adapting to changes in the environment. Those SI techniques based on the behaviour of real ant colonies used to solve discrete optimization problems are classed as ant colony optimization (ACO) techniques [3]. These have been successfully applied to a large number of difficult combinatorial problems like the quadratic assignment [15] and the traveling salesman [8] problems, to routing in telecommunications networks, scheduling, and other problems. This method is particularly attractive for feature selection as there seems to be no heuristic that can guide search to the optimal minimal subset every time. Additionally, it can be the case that ants discover the best feature combinations as they proceed throughout the search space. This paper investigates how ACO may be applied to the difficult problem of finding optimal feature subsets.

The rest of this paper is structured as follows. Section 2 describes the theory of fuzzy-rough set data reduction, with the aid of a simple example. Section 3 introduces the main concepts in ACO and details how this may be applied to the problem of feature selection in general, and fuzzy-rough feature selection

in particular. Section 4 describes the experimentation carried out on the real problem case of complex system monitoring and presents the results. Section 5 concludes the paper, and proposes further work in this area.

2. Fuzzy-rough data reduction

The crisp rough set-based feature selection (RSFS) process described in [5] can only operate effectively with datasets containing discrete values. As most datasets contain real-valued features, it is necessary to perform a discretization step beforehand. This is typically implemented by standard fuzzification techniques [16]. However, membership degrees of feature values to fuzzy sets are not exploited in the process of dimensionality reduction. By using *fuzzy-rough* sets [10,17,22], it is possible to use this information to better guide feature selection.

2.1. 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 [10]. For typical RSFS applications, 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. The usual properties of reflexivity ($\mu_S(x, x) = 1$), symmetry ($\mu_S(x, y) = \mu_S(y, x)$) and transitivity ($\mu_S(x, z) \geqslant \mu_S(x, y) \wedge \mu_S(y, z)$) hold.

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

$$\mu_{[x]_S}(y) = \mu_S(x, y). \tag{1}$$

The following axioms should hold for a fuzzy equivalence class F [11]:

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

The first axiom corresponds to the requirement that an equivalence class is non-empty. The second axiom states that elements in y's neighbourhood 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 non-fuzzy.

The family of normal fuzzy sets produced by a fuzzy partitioning of the universe of discourse can play the role of fuzzy equivalence classes [10]. Consider the crisp partitioning $N_a = \{1, 3, 6\}$, $Z_a = \{2, 4, 5\}$ over the universe $\mathbb{U} = N_a \cup Z_a$. This contains two equivalence classes (N_a and Z_a) 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 N_a, 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]. Equivalence classes are not restricted to crisp partitions only; fuzzy partitions are equally acceptable.

2.2. Fuzzy lower and upper approximations

From the literature, the fuzzy P-lower and P-upper approximations are defined as [10]

$$\mu_{\underline{P}X}(F_i) = inf_x \, max\{1 - \mu_{F_i}(x), \mu_X(x)\} \quad \forall i, \tag{2}$$

$$\mu_{\overline{P}X}(F_i) = sup_x \, min\{\mu_{F_i}(x), \mu_X(x)\} \quad \forall i, \tag{3}$$

where F_i denotes a fuzzy equivalence class belonging to $\mathbb{U}/P$ which in turn stands for the partition of $\mathbb{U}$ with respect to a given subset P of features.

For an individual feature, a, the partition of the universe by $\{a\}$ (denoted $\mathbb{U}/IND(\{a\})$) is considered to be the set of those fuzzy equivalence classes for that feature. For example, if the two fuzzy sets N_a and Z_a are generated for feature a during fuzzification, the partition $\mathbb{U}/IND(\{a\}) = \{N_a, Z_a\}$. If the fuzzy-rough reduction process is to be useful, it must be able to deal with multiple features, finding the dependency between various subsets of the original feature set. For example, it may be necessary to be able to determine the degree of dependency of the decision feature(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 features 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,

$$\mathbb{U}/P = \otimes\{a \in P : \mathbb{U}/IND(\{a\})\}. \tag{4}$$

For example, if $P = \{a, b\}$, $\mathbb{U}/IND(\{a\}) = \{N_a, Z_a\}$ and $\mathbb{U}/IND(\{b\}) = \{N_b, Z_b\}$, then

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

Note that although the universe of discourse in feature reduction is finite, this is not the case in general, hence the use of *sup* and *inf* above. 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 redefined as [14]

$$\mu_{\underline{P}X}(x) = \sup_{F \in \mathbb{U}/P} min(\mu_F(x), \inf_{y \in \mathbb{U}} max\{1 - \mu_F(y), \mu_X(y)\}), \tag{5}$$

$$\mu_{\overline{P}X}(x) = \sup_{F \in \mathbb{U}/P} min(\mu_F(x), \sup_{y \in \mathbb{U}} min\{\mu_F(y), \mu_X(y)\}). \tag{6}$$

In implementation, not all $y \in \mathbb{U}$ are needed to be considered—only those where $\mu_F(y)$ is non-zero, 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.

Each set in $\mathbb{U}/P$ denotes an equivalence class. 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 $F_i, i = 1, 2, \ldots, n$:

$$\mu_{F_1 \cap \ldots \cap F_n}(x) = min(\mu_{F_1}(x), \mu_{F_2}(x), \ldots, \mu_{F_n}(x)). \tag{7}$$

2.3. Fuzzy-rough reduction process

FRFS builds on the notion of the fuzzy lower approximation to enable reduction of datasets containing real-valued features. As will be shown, the process becomes identical to the crisp approach when dealing with nominal well-defined features.

The crisp positive region in the standard RST 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

$$\mu_{\mathrm{POS}_P(Q)}(x) = \sup_{X \in \mathbb{U}/Q} \mu_{\underline{P}X}(x). \tag{8}$$

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:

$$\gamma'_P(Q) = \frac{|\mu_{\mathrm{POS}_P(Q)}(x)|}{|\mathbb{U}|} = \frac{\sum_{x \in \mathbb{U}} \mu_{\mathrm{POS}_P(Q)}(x)}{|\mathbb{U}|}. \tag{9}$$

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

A problem may arise when this approach is compared to the crisp approach. In conventional rough set-based feature selection, a reduct is defined as a subset R of the features which have the same information content as the full feature 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 dataset contains no contradictory information. 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 a set of decision features D upon the full feature set and use this as the denominator rather than $|\mathbb{U}|$ (for normalization), allowing γ' to reach 1. With these issues in mind, a new QuickReduct algorithm, based on the crisp version [5], has been developed as given in Fig. 1. It employs the new dependency function γ' to choose which features to add to the current reduct candidate. The algorithm terminates when the addition of any remaining feature does not increase the dependency. As with the original algorithm, for a dimensionality of n, the worst case dataset will result in $(n^2 + n)/2$ evaluations of the dependency function. However, as fuzzy-rough set-based feature selection is used for dimensionality reduction prior to any involvement of the system which will employ those features belonging to the resultant reduct, this operation has no negative impact upon the run-time efficiency of the system.

Note that it is also possible to reverse the search process; that is, start with the full set of features and incrementally remove the least informative features. This process continues until no more features can be removed without reducing the total number of discernible objects in the dataset. This approach is less suitable for data reduction when the dataset's dimensionality is very large.

FRQUICKREDUCT(C,D).
C, the set of all conditional features;
D, the set of decision features.

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

Fig. 1. The fuzzy-rough QUICKREDUCT algorithm.

Table 1
Example dataset

Object	A			B			C		Plan		
	$A1$	$A2$	$A3$	$B1$	$B2$	$B3$	$C1$	$C2$	X	Y	Z
0	0.3	0.7	0.0	0.2	0.7	0.1	0.3	0.7	0.1	0.9	0.0
1	1.0	0.0	0.0	1.0	0.0	0.0	0.7	0.3	0.8	0.2	0.0
2	0.0	0.3	0.7	0.0	0.7	0.3	0.6	0.4	0.0	0.2	0.8
3	0.8	0.2	0.0	0.0	0.7	0.3	0.2	0.8	0.6	0.3	0.1
4	0.5	0.5	0.0	1.0	0.0	0.0	0.0	1.0	0.6	0.8	0.0
5	0.0	0.2	0.8	0.0	1.0	0.0	0.0	1.0	0.0	0.7	0.3
6	1.0	0.0	0.0	0.7	0.3	0.0	0.2	0.8	0.7	0.4	0.0
7	0.1	0.8	0.1	0.0	0.9	0.1	0.7	0.3	0.0	0.0	1.0
8	0.3	0.7	0.0	0.9	0.1	0.0	1.0	0.0	0.0	0.0	1.0

2.4. A worked example

Using the fuzzy-rough QUICKREDUCT algorithm, Table 1 can be reduced in size. First of all the lower approximations need to be determined. Consider the first feature in the dataset. Setting $P = \{A\}$ produces the fuzzy partitioning $\mathbb{U}/P = \{A1, A2, A3\}$, and setting $Q = \{Plan\}$ produces the fuzzy partitioning $\mathbb{U}/Q = \{X, Y, Z\}$. To determine the fuzzy P-lower approximation of Plan X ($\mu_{\underline{P}X}(x)$), each $F \in \mathbb{U}/P$ must be considered. For $F = A1$:

$$min(\mu_{A1}(x), \inf_{y\in\mathbb{U}} max\{1 - \mu_{A1}(y), \mu_X(y)\}) = min(\mu_{A1}(x), 0.6).$$

Similarly, for $F = A2$, $min(\mu_{A2}(x), 0.3)$ and $F = A3$, $min(\mu_{A3}(x), 0.0)$. To calculate the extent to which an object x in the dataset belongs to the fuzzy P-lower approximation of X, the union of these values is calculated. For example, object 0 belongs to $\underline{P}X$ with a membership of:

$$sup\{min(\mu_{A1}(0), 0.6), min(\mu_{A2}(0), 0.3), min(\mu_{A3}(0), 0.0)\} = 0.3.$$

Likewise, for Y and Z:

$$\mu_{\underline{P}Y}(0) = 0.2 \; \mu_{\underline{P}Z}(0) = 0.3.$$

The extent to which object 0 belongs to the fuzzy positive region can be determined by considering the union of fuzzy P-lower approximations:

$$\mu_{POS_P(Q)}(0) = \sup_{S \in \mathbb{U}/Q} \mu_{\underline{P}S}(0) = 0.3.$$

Similarly, for the remaining objects,

$$\begin{aligned} &\mu_{\mathrm{POS}_P(Q)}(1) = 0.6, \quad \mu_{\mathrm{POS}_P(Q)}(2) = 0.3, \\ &\mu_{\mathrm{POS}_P(Q)}(3) = 0.6, \quad \mu_{\mathrm{POS}_P(Q)}(4) = 0.5, \\ &\mu_{\mathrm{POS}_P(Q)}(5) = 0.3, \quad \mu_{\mathrm{POS}_P(Q)}(6) = 0.6, \\ &\mu_{\mathrm{POS}_P(Q)}(7) = 0.3, \quad \mu_{\mathrm{POS}_P(Q)}(8) = 0.3. \end{aligned}$$

Using these values, the degree of dependency of Q on $P = \{A\}$ can be calculated:

$$\gamma'_P(Q) = \frac{\sum_{x \in \mathbb{U}} \mu_{\mathrm{POS}_P(Q)}(x)}{|0, 1, 2, 3, 4, 5, 6, 7, 8|} = 3.8/9.$$

The fuzzy-rough QuickReduct algorithm uses this process to evaluate subsets of features in an incremental fashion. The algorithm starts with an empty set and considers the addition of each individual feature:

$$\gamma'_{\{A\}}(Q) = 3.8/9, \quad \gamma'_{\{B\}}(Q) = 2.1/9, \quad \gamma'_{\{C\}}(Q) = 2.7/9.$$

As feature A causes the greatest increase in dependency degree, it is added to the reduct candidate and the search progresses:

$$\gamma'_{\{A,B\}}(Q) = 4.0/9, \quad \gamma'_{\{A,C\}}(Q) = 5.7/9.$$

Here, C is added to the reduct candidate as the dependency is increased. There is only one feature addition to be checked at the next stage, namely

$$\gamma'_{\{A,B,C\}}(Q) = 5.7/9.$$

This causes no dependency increase, resulting in the algorithm terminating and outputting the reduct $\{A, C\}$. Hence, the original dataset can be reduced to these features with minimal information loss (according to the algorithm).

3. ACO for feature selection

The ability of real ants to find shortest routes is mainly due to their depositing of pheromone as they travel; each ant probabilistically prefers to follow a direction rich in this chemical. The pheromone decays over time, resulting in much less pheromone on less popular paths. Given that over time the shortest route will have the higher rate of ant traversal, this path will be reinforced and the others diminished until all ants follow the same, shortest path (the "system" has converged to a single solution). It is also possible that there are many equally short paths. In this situation, the rates of ant traversal over the short paths will be roughly the same, resulting in these paths being maintained while others are ignored. Additionally, if a sudden change to the environment occurs (e.g. a large obstacle appears on the shortest path), the ACO system can respond to this and will eventually converge to a new solution.

3.1. Premise of application of ACO algorithms

In general, an ACO algorithm can be applied to any combinatorial problem as far as it is possible to define:

Appropriate problem representation: The problem can be described as a graph with a set of nodes and edges between nodes.

Heuristic desirability (η) *of edges*: A suitable heuristic measure of the "goodness" of paths from one node to every other connected node in the graph.

Construction of feasible solutions: A mechanism must be in place whereby possible solutions are efficiently created. This requires the definition of a suitable traversal stopping criterion to stop path construction when a solution has been reached.

Pheromone updating rule: A suitable method of updating the pheromone levels on edges is required with a corresponding evaporation rule, typically involving the selection of the n best ants and updating the paths they chose.

Probabilistic transition rule: The rule that determines the probability of an ant traversing from one node in the graph to the next.

The feature selection task may be reformulated into an ACO-suitable problem. ACO requires a problem to be represented as a graph—here nodes represent features, with the edges between them denoting the choice of the next feature. The search for the optimal feature subset is then an ant traversal through the graph where a minimum number of nodes are visited that satisfies the traversal stopping criterion. Fig. 2 illustrates this setup—the ant is currently at node a and has a choice of which feature to add next to its path (dotted lines). It chooses feature b next based on the transition rule, then c and then d. Upon arrival at d, the current subset $\{a, b, c, d\}$ is determined to satisfy the traversal stopping criterion (e.g. a suitably high classification accuracy has been achieved with this subset). The ant terminates its traversal and outputs this feature subset as a candidate for data reduction.

A suitable heuristic desirability of traversing between features could be any subset evaluation function—for example, the fuzzy-rough set dependency measure. This measure gives an indication of which features are more informative given the currently selected subset. The heuristic desirability of traversal and edge pheromone levels are combined to form the so-called probabilistic transition rule [3], denoting the

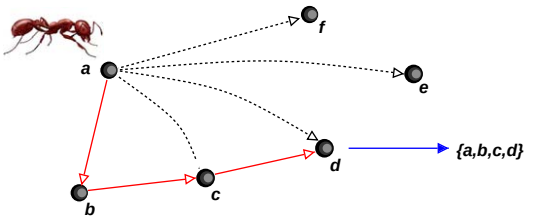

Fig. 2. ACO problem representation for FS.

probability of an ant at feature i choosing to travel to feature j at time t:

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum_{l \in J_i^k}[\tau_{il}(t)]^{\alpha}.[\eta_{il}]^{\beta}}, \tag{10}$$

where k is the number of ants, J_i^k is the set of ant k's unvisited features, η_{ij} is the heuristic desirability of choosing feature j when at feature i and $\tau_{ij}(t)$ is the amount of virtual pheromone on edge (i, j). The choice of parameters α and β is determined experimentally. Several parameter values are chosen in the range [0, 1] and evaluated by experimentation.

Depending on how optimality is defined for the particular application, the pheromone may be updated accordingly. For instance, subset minimality and "goodness" are two key factors so the pheromone update must be proportional to "goodness" and inversely proportional to size. There is also the possibility of allowing the removal of features here. If feature h has been selected already, an alternative transition rule may be applied to determine the probability of removing this attribute. However, this is an extension of the ant-based feature selection approach and is not required for its operation.

The time complexity of the ant-based approach to feature selection is $O(IAk)$, where I is the number of iterations, A the number of original features, and k the number of ants. This can be seen from Fig. 3. In the worst case, each ant selects all the features. As the heuristic is evaluated after each feature is added to the reduct candidate, this will result in A evaluations per ant. After one iteration in this scenario, Ak evaluations will have been performed. After I iterations, the heuristic will be evaluated IAk times.

3.2. ACO for fuzzy-rough feature selection

The overall process of ACO feature selection can be seen in Fig. 3. It begins by generating a number of ants, k, which are then placed randomly on the graph (i.e. each ant starts with one random feature). Alternatively, the number of ants to place on the graph may be set equal to the number of features within the data; each ant starts path construction at a different feature. From these initial positions, they traverse edges probabilistically until a traversal stopping criterion is satisfied. The resulting subsets are gathered and then evaluated. If an optimal subset has been found or the algorithm has executed a certain number of times, then the process halts and outputs the best feature subset encountered. If neither condition holds, then the pheromone is updated, a new set of ants are created and the process iterates once more.

To tailor this mechanism to find fuzzy-rough set reducts, it is necessary to use the dependency measure given in Eq. (9) as the stopping criterion. This means that an ant will stop building its feature subset when the dependency of the subset reaches the maximum for the dataset (the value 1 for consistent datasets). The

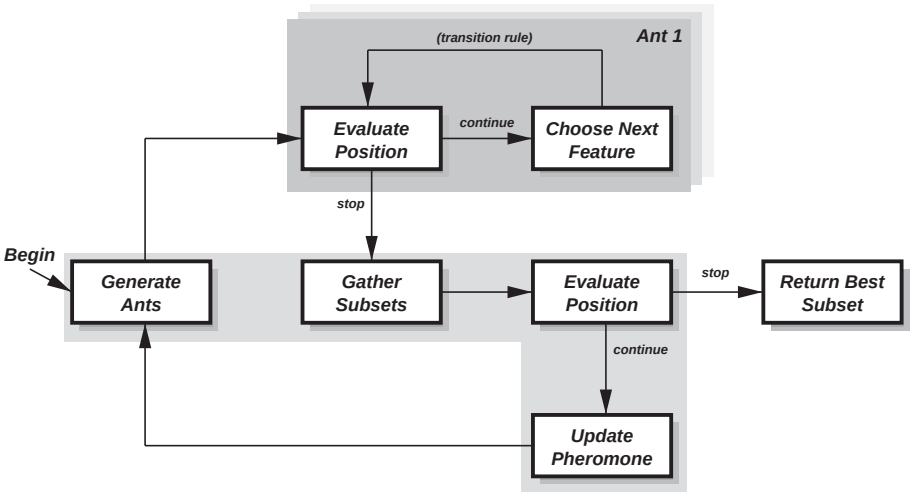

Fig. 3. ACO-based feature selection overview.

dependency function may also be chosen as the heuristic desirability measure, but this is not necessary. In fact, it may be of more use to employ a non-rough set related heuristic for this purpose to avoid the pitfalls of a QUICKREDUCT style search. By using an alternative measure such as an entropy-based heuristic [20], the method may avoid feature combinations that may mislead the rough set-based heuristic. Again, the time complexity of this fuzzy-rough ant-based method will be the same as that mentioned earlier, $O(IAk)$.

The pheromone on each edge is updated according to the following formula:

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \Delta\tau_{ij}(t), \tag{11}$$

where

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{n} (\gamma'(S^k)/|S^k|). \tag{12}$$

This is the case if the edge (i, j) has been traversed; $\Delta\tau_{ij}(t)$ is 0 otherwise. The value ρ is a decay constant used to simulate the evaporation of the pheromone, S^k is the feature subset found by ant k. The pheromone is updated according to both the fuzzy-rough measure of the "goodness" of the ant's feature subset (γ') and the size of the subset itself. By this definition, all ants update the pheromone. Alternative strategies may be used for this, such as allowing only the ants with the best feature subsets to proportionally increase the pheromone. These are, however, beyond the scope of this paper.

4. Experimentation

To show the utility of fuzzy-rough feature selection (FRFS) and to compare the hill-climbing and ant-based fuzzy-rough approaches, the two methods are applied as pre-processors within a complex

systems monitoring application. Both methods preserve the semantics of the surviving features after removing redundant ones. This is essential in satisfying the requirement of user readability of the generated knowledge model, as well as ensuring the understandability of the pattern classification process.

4.1. Test domain

In order to evaluate the fuzzy-rough approaches and to illustrate its domain-independence, a challenging test dataset was chosen, namely the Water Treatment Plant Database [2]. The dataset itself is a set of historical data charted over 521 days, with 38 different input features measured daily. Each day is classified into one of thirteen categories depending on the operational status of the plant. However, these can be collapsed into just two or three categories (i.e. *Normal* and *Faulty*, or *OK*, *Good* and *Faulty*) for plant monitoring purposes as many classifications reflect similar performance. Because of the efficiency of the actual plant the measurements were taken from, all faults appear for short periods (usually single days) and are dealt with immediately. This does not allow for a lot of training examples of faults, which is a clear drawback if a monitoring system is to be produced. Collapsing 13 categories into 2 or 3 classes helps reduce this difficulty for the present application. Note that this dataset has been utilized in many previous studies, including that reported in [21] (to illustrate the effectiveness of applying crisp RSFS as a pre-processing step to rule induction).

It is likely that not all of the 38 input features are required to determine the status of the plant, hence the dimensionality reduction step. However, choosing the most informative features is a difficult task as there will be many dependencies between subsets of features. There is also a monetary cost involved in monitoring these inputs, so it is desirable to reduce this number.

Note that the original monitoring system (Fig. 4) developed in [21] consisted of several modules; it is this modular structure that allows the FRFS techniques to replace the existing crisp method. Originally, a precategorization step preceded feature selection where feature values were quantized. To reduce potential loss of information, the original use of just the dominant symbolic labels of the discretized fuzzy terms is now replaced by a fuzzification procedure. This leaves the underlying feature values unchanged but generates a series of fuzzy sets for each feature. These sets are generated entirely from the data while exploiting the statistical properties attached to the dataset (in keeping with the rough set ideology in that the dependence of learning upon information provided outside of the training dataset is minimized). This module may be replaced by alternative fuzzifiers, or expert-defined fuzzification if available. Based on these fuzzy sets and the original real-valued dataset, the feature selection module calculates a reduct and reduces the dataset accordingly. Finally, rule induction is performed on the reduced dataset. For this set of experiments, the decision tree method C4.5 [20] is used for induction and the learned rules for classification.

4.2. Experimental results

This section presents the results from the various comparative studies. The first set of experiments compares the hill-climbing and ant-based fuzzy-rough methods. An investigation into another feature selector based on the entropy measure is then presented. This is followed by comparisons with a transformation-based approach, PCA.

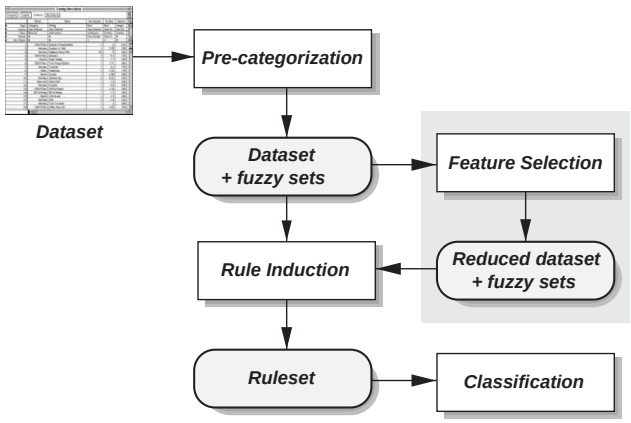

Fig. 4. Modular decomposition of the implemented system.

Table 2
Results for the 2-class dataset

Method	Attributes	γ' value	Training error (%)	Testing error (%)
Unreduced	38	—	1.5	19.1
FRFS	10	0.58783	10.8	25.2
antFRFS	9.55	0.58899	6.5	22.1

4.2.1. Comparison of fuzzy-rough methods

Three sets of experiments were carried out on both the (collapsed) 2-class and 3-class datasets. The first bypasses the feature selection part of the system, using the original water treatment dataset as input to C4.5, with all 38 conditional attributes. The second method employs FRFS to perform the feature selection before induction is carried out. The third uses the ant-based method described in Section 3 (antFRFS) to perform feature selection over a number of runs, and the results averaged.

The results for the 2-class dataset can be seen in Table 2. Both FRFS and antFRFS significantly reduce the number of original attributes with antFRFS producing the greatest data reduction on average. As well as generating smaller reducts, antFRFS finds reducts of a higher quality according to the fuzzy-rough dependency measure. This higher quality is reflected in the resulting classification errors for both the training and testing datasets, with antFRFS outperforming FRFS.

Table 3 shows the results for the 3-class dataset experimentation. The hill-climbing fuzzy-rough method chooses 11 out of the original 38 features. The ant-based method chooses fewer attributes on average, however this is at the cost of a lower dependency measure for the generated reducts. Again the effect of this can be seen in the classification errors, with FRFS performing slightly better than antFRFS. For both fuzzy methods, the small drop in classification accuracy as a result of feature selection is acceptable.

By selecting a good feature subset from data it is usually expected that the applied learning method should benefit, resulting in an improvement in results. However, when the original training (and test) data is very noisy, selected features may not necessarily be able to reflect all the information contained within the original entire feature set. As a result of removing less informative features, partial useful information

Table 3
Results for the 3-class dataset

Method	Attributes	γ' value	Training error (%)	Testing error (%)
Unreduced	38	—	2.1	16.8
FRFS	11	0.59479	2.8	19.1
antFRFS	9.09	0.58931	5.2	19.8

Table 4
Results for the three selection methods

Approach	No. of classes	No. of features	Training error (%)	Testing error (%)
FRFS	2	10	10.8	25.2
antFRFS	2	9.55	6.5	22.1
Entropy	2	13	2.3	19.8
FRFS	3	11	2.8	19.1
antFRFS	3	9.09	5.2	19.8
Entropy	3	14	1.8	19.1

may be lost. The goal of selection methods in this situation is to minimize this loss, while reducing the number of features to the greatest extent. Therefore, it is not surprising that the classification performance for this challenging dataset can decrease upon data reduction, as shown in Table 3. However, the impact of feature selection can have different effects on different classifiers. With the use of an alternative classifier in Section 4.2.4, performance can be seen to improve for the test data.

4.2.2. Comparison with entropy-based feature selection

To support the study of the performance of the fuzzy-rough methods for use as pre-processors to rule induction, a conventional entropy-based technique is used for comparison. This approach utilizes the entropy heuristic typically employed by machine learning techniques such as C4.5 [20]. Those features that provide the most gain in information are selected. A summary of the results of this comparison can be seen in Table 4.

For both the 2-class and 3-class datasets, FRFS and antFRFS select at least three fewer features than the entropy-based method. However, the entropy-based method outperforms the other two feature selectors with the resulting C4.5 classification accuracies. This is probably due to the fact that C4.5 uses exactly the same entropy measure in generating decision trees. In this case, the entropy-based measure will favour those attributes that will be the most influential in the decision tree generation process.

4.2.3. Comparison with the use of PCA

The effect of using a different dimensionality reduction technique, namely PCA [7], is also investigated. Here, PCA is applied to the dataset and the first n principal components are used. A range of values is chosen for n to investigate how the performance varies with dimensionality. As PCA irreversibly destroys the underlying dataset semantics, the resulting decision trees are not human-comprehensible nor directly measurable but may still provide useful automatic classifications of new data. Table 5 shows the results from applying PCA to the datasets.

Table 5
Results for the 2-class and 3-class datasets using PCA

Error	Class	No. of features								
		5	6	7	8	9	**10**	11	12	13
Training (%)	2	20.0	20.0	20.0	20.0	19.7	**19.7**	19.7	19.2	17.9
Testing (%)	2	27.5	27.5	27.5	27.5	26.7	**26.7**	26.7	64.9	65.6
Training (%)	3	26.4	26.4	26.4	26.4	26.4	**24.1**	24.1	24.1	23.6
Testing (%)	3	19.1	19.1	19.1	19.1	19.1	**19.1**	19.1	19.1	19.8

Table 6
Results for the 2-class and 3-class datasets using SMO

Approach	No. of classes	No. of features	Training error (%)	Testing error (%)
Unreduced	2	38	20.0	28.2
FRFS	2	10	20.0	27.5
antFRFS	2	9.55	20.0	27.5
Unreduced	3	38	25.4	19.1
FRFS	3	11	26.4	19.8
antFRFS	3	9.09	26.4	19.1

Both antFRFS and FRFS significantly outperform PCA on the 2-class dataset. Of particular interest is when 10 principal components are used as this is roughly the same number chosen by antFRFS and FRFS. The resulting error for PCA is 19.7% for the training data and 26.7% for the test data. For antFRFS the errors were 6.5% (training) and 22.1% (testing), and for FRFS 10.8% (training) and 25.2% (testing). In the 3-class dataset experimentation, both fuzzy-rough methods produce much lower classification errors than PCA for the training data. For the test data, the performance is about the same, with PCA producing a slightly lower error than antFRFS on the whole.

4.2.4. Comparison with the use of a support vector classifier

A possible limitation of C4.5 in this context is that it performs a degree of feature selection itself during the induction process. The resulting decision trees do not necessarily contain all the features present in the original training data. As a result of this, it is beneficial to evaluate the use of an alternative classifier that uses all the given features. For this purpose, a support vector classifier is used, trained by the sequential minimal optimization (SMO) algorithm [19]. The results of the application of this classifier can be found in Table 6.

For the 2-class dataset, the training error for both FRFS and antFRFS is the same as that of the unreduced approach. However, this is with significantly fewer attributes. Additionally, the resulting testing error *is* reduced with these feature selection methods. With the more challenging 3-class problem, the training errors are slightly worse (as seen with the C4.5 analysis). The antFRFS method performs better than FRFS for the test data and is equal to the unreduced method, again using fewer features.

5. Conclusion

This paper has 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. When minimal data reductions are required, other search mechanisms must be employed. Although these methods also cannot ensure optimality, they provide a means by which the best feature subsets might be found. This paper has presented a new method for feature selection based on ant colony optimization for this purpose. This was applied to the problem of fuzzy-rough dimensionality reduction, with promising results. Unlike semantics-destroying approaches such as PCA, this approach maintains the underlying semantics of the feature set, thereby ensuring that the resulting models are interpretable and the inference explainable.

In all experimental studies there has been no attempt to optimize the fuzzifications or the classifiers employed. It can be expected that the results obtained with optimization would be even better than those already observed. The generality of this approach should enable it to be applied to other domains. The decision trees generated by the induction method was not processed by any post-processing tools so as to allow its behaviour and capabilities to be revealed fully. By enhancing the induced decision tree through post-processing, performance can be expected to improve. Additionally, fuzzy or alternatively other crisp rule induction algorithms [4,9] may be used which should benefit more from a feature selection method that uses the fuzzification information for data reduction. The current decision tree method may be easily replaced due to the modularity of the system.

Work is being carried out on a fuzzified dependency function [13]. Ordinarily, the dependency function returns values for sets of features in the range [0,1]; the fuzzy dependency function will return qualitative fuzzy labels for use in the fuzzy-rough QUICKREDUCT algorithm. With this mechanism in place, several features may be chosen at one time according to their labels, speeding up the feature selection process. Additionally, research is being carried out into the potential utility of *fuzzy reducts*, which would allow features to have a varying possibility of becoming a member of the resultant reduct. Further work also includes broadening the comparative studies to include comparisons with other feature selection and dimensionality reduction techniques.

Acknowledgements

This work is partly funded by the UK EPSRC Grant 00317404. The authors are very grateful to David Robertson and the other members of the Advanced Knowledge Technologies [1] team at Edinburgh.

References

[1] Advanced Knowledge Technologies homepage: http://www.aktors.org/
[2] C.L. Blake, C.J. Merz, UCI Repository of machine learning databases. Irvine, University of California, 1998, http://www.ics.uci.edu/~mlearn/.
[3] E. Bonabeau, M. Dorigo, G. Theraulez, Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press Inc., NY, USA, 1999.
[4] S. Chen, S.L. Lee, C. Lee, A new method for generating fuzzy rules from numerical data for handling classification problems, Applied Artificial Intelligence 15 (7) (2001) 645–664.

[5] A. Chouchoulas, Q. Shen, Rough set-aided keyword reduction for text categorisation, Appl. Artif. Intell. 15 (9) (2001) 843–873.
[6] M. Dash, H. Liu, Feature selection for classification, Intell. Data Anal. 1 (3) (1997) 131–156.
[7] P. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach, Prentice-Hall, Englewood Cliffs, NJ, 1982.
[8] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: optimization by a colony of cooperating agents, IEEE Trans. Systems Man Cybern. Part B 26 (1) (1996) 29–41.
[9] D. Dubois, E. Hüllermeier, H. Prade, On the representation of fuzzy rules in terms of crisp rules, Inform. Sci. 151 (2003) 301–326.
[10] D. Dubois, H. Prade, Putting rough sets and fuzzy sets together, in: R. Slowinsk (Ed.), Intelligent Decision Support, Kluwer Academic Publishers, Dordrecht, 1992, pp. 203–232.
[11] U. Höhle, Quotients with respect to similarity relations, Fuzzy Sets and Systems 27 (1988) 31–44.
[12] R. Jensen, Q. Shen, Finding rough set reducts with ant colony optimization, in: Proc. 2003 UK Workshop on Computational Intelligence, 2003, pp. 15–22.
[13] R. Jensen, Q. Shen, Using fuzzy dependency-guided attribute grouping in feature selection, in: Proc. 9th Internat. Conf. on Rough Sets, 2003, pp. 250–254.
[14] R. Jensen, Q. Shen, Fuzzy-rough attribute reduction with application to web categorization, Fuzzy Sets and Systems 141 (3) (2004) 469–485.
[15] V. Maniezzo, A. Colorni, The ant system applied to the quadratic assignment problem, Knowledge Data Eng. 11 (5) (1999) 769–778.
[16] J.G. Marin-Blázquez, Q. Shen, From approximative to descriptive fuzzy classifiers, IEEE Trans. Fuzzy Systems 10 (4) (2002) 484–497.
[17] S.K. Pal, A. Skowron (Eds.), Rough-Fuzzy Hybridization: A New Trend in Decision Making, Springer, Singapore, 1999.
[18] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishing, Dordrecht, 1991.
[19] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schlkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge, MA, 1998.
[20] J.R. Quinlan, C4.5: Programs for Machine Learning, The Morgan Kaufmann Series in Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
[21] Q. Shen, A. Chouchoulas, A modular approach to generating fuzzy rules with reduced attributes for the monitoring of complex systems, Engineering Appl. Artif. Intell. 13 (3) (2000) 263–278.
[22] R. Slowinski (Ed.), Intelligent Decision Support, Kluwer Academic Publishers, Dordrecht, 1992.
[23] L.A. Zadeh, Fuzzy sets, Inform. Control 8 (1965) 338–353.